

كتاب تقن 102



تم تحميل هذا الملف من موقع مناهج مملكة البحرين

موقع المناهج ← مناهج مملكة البحرين ← الصف الأول الثانوي ← علوم وتقانة ← الفصل الثاني ← ملفات متنوعة ← الملف

تاريخ إضافة الملف على موقع المناهج: 2026-06-08 12:45:10

ملفات اكتب للمعلم اكتب للطالب ا اختبارات الكترونية ا اختبارات ا حلول ا عروض بوربوينت ا أوراق عمل
منهج انجليزي ا ملخصات وتقارير ا مذكرات وبنوك ا الامتحان النهائي ا للمدرس

المزيد من مادة
علوم وتقانة:

التواصل الاجتماعي بحسب الصف الأول الثانوي



صفحة مناهج مملكة
البحرين على
فيسبوك

الرياضيات

اللغة الانجليزية

اللغة العربية

التربية الاسلامية

المواد على تلغرام

المزيد من الملفات بحسب الصف الأول الثانوي والمادة علوم وتقانة في الفصل الثاني

ملخص شامل لمقرر البرمجة بلغة Python (تقن 108)

1

مراجعة و حل مذكرة تقن 107 للفصل الأول

2

نموذج إجابات امتحانات نهاية الفصل الثاني

3

أسئلة امتحانات سابقة

4

كتاب تقن 102

5

الدرس الأول

الأنظمة العددية

Numbering systems

أهداف الدرس



يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- ما هي الأنظمة العددية.
- 2- تعريف نظام العدّ العشري.
- 3- تعريف نظام العدّ الثنائي.
- 4- تفسير العلاقة بين لغة الحاسوب ونظام العدّ الثنائي.
- 5- تحويل عدد ما من النظام العشري إلى النظام الثنائي.
- 6- تحويل عدد ما من النظام الثنائي إلى النظام العشري.

1- ما هي الأنظمة العددية Numbering system

هي أنظمة عدّ أو ترقيم تعتمد أولاً على مجموعة محدّدة من الرموز لتمثيل الأعداد وثانياً على منهجية معيّنة لكتابتها وعرضها. يحتلّ كلّ رمز من رموز النظام العددي موضعاً يمثّل قيمته.

مثال: الرمز 1 ضمن العدد 10 أقلّ قيمة منه ضمن العدد 100

2- تعريف نظام العدّ العشري Decimal numbering system

من أهمّ الأنظمة العددية الذي يعتمد على الرموز من 0 إلى 9 لتمثيل الأعداد العشرية نسبة إلى العدد 10 الذي يمثّل بدوره عدد الرموز من 0 إلى 9. وهو النظام العددي الأكثر شيوعاً والذي نعتمده في حساباتنا اليومية وفي أنظمتنا الاقتصادية وإدارتنا التعليمية ومختلف المجالات الأخرى.

3- تعريف نظام العدّ الثنائي Binary numbering system

هو نظام عدّ يعتمد على الرموز من 0 إلى 1 لتمثيل الأعداد الثنائية نسبة إلى العدد 2 الذي يمثّل بدوره عدد الرموز من 0 إلى 1.

4- العلاقة بين لغة الحاسوب ونظام العدّ الثنائي

لو لم يكن هذا النظام لما كان بإمكانك تنفيذ أيّ من العمليات الإلكترونية التي تقوم بها يومياً، بدءاً من هذا الكتاب الذي تقرأه حالياً، فهو يعتبر النظام الخاص بلغة الآلة.

يعمل الحاسوب بالطاقة الكهربائية ما يعني أنّه يتعامل مع الإشارات فإمّا أن تكون هناك إشارة كهربائية ويرمز إليها بإحدى القيم الآتية: (1 || ON || TRUE || YES) أو عدم وجود إشارة كهربائية ويرمز إلى ذلك بإحدى القيم الآتية: (0 || OFF || FALSE || NO)، يجدر الإشارة إلى أنّ أكثر القيم استخداماً وشيوعاً هي (1-0).

ليستطيع جهاز الحاسوب فهم البيانات المدخلة سواء كانت أعداداً ونصوصاً أو صوراً وأصواتاً ومقاطع فيديو أو غيرها من البيانات، يجب أن تحوّل أولاً إلى اللغة الثنائية (1-0) ومن ثمّ معالجتها ثمّ عرض/تنفيذ نتائجها من خلال أجهزة الإخراج.

مثال:

طباعة مستند: الإدخال (مسح المستند) - المعالجة (تحويله إلى مستند رقمي (1001...)) - الإخراج (الطباعة)

5- تحويل عدد ما من النظام العشري إلى النظام الثنائي

هناك طريقتان لتحويل عدد عشري إلى عدد ثنائي.

الطريقة الأولى: طريقة القسمة المتكررة على الرقم 2

مثال: لتحويل العدد $_{10}(47)$ إلى $_2(???)$ أتبع الخطوات الآتية:

$$\begin{array}{rcl} 23 = 2 \div 47 & \text{الباقي} & 1 \\ 11 = 2 \div 23 & \text{الباقي} & 1 \\ 5 = 2 \div 11 & \text{الباقي} & 1 \\ 2 = 2 \div 5 & \text{الباقي} & 1 \\ 1 = 2 \div 2 & \text{الباقي} & 0 \end{array}$$

إعادة لعملية القسمة المتكررة من خلال الجدول الآتي:

2	5	11	23	47	القاسم
2	2	2	2	2	المقسوم عليه
1	2	5	11	23	النتيجة
0	1	1	1	1	الباقي

ألاحظ:

- 1- تتوقف عملية القسمة عندما تكون نتيجة القسمة أصغر من المقسوم عليه.
- 2- أول باقي من عملية القسمة يكون موقعه أقصى اليمين من العدد الثنائي ثم بقية البواقي على يساره.
- 3- آخر نتيجة من عملية القسمة يكون موقعه أقصى اليسار في العدد الثنائي.

تكون النتيجة كالآتي: $_{10}(47) = _2(101111)$

الطريقة الثانية: جدول الأعداد العشرية قوة 2

.....	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
.....	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
.....								1	0	1	1	1	1

لإيجاد نتيجة تحويل العدد (47)₁₀ إلى عدد ثنائي أتبع الخطوات الآتية:

1- أبدأ بوضع الرقم 1 في الخانة أسفل أقرب عدد عشري للعدد المطلوب تحويله. في حالتنا هذه أقرب عدد عشري للعدد 47 ضمن الجدول أعلاه هو 32، ثم أضع ضمن الخانات على يمينه أيضا الرقم 1 تحت الأعداد العشرية المناسبة بحيث يكون مجموعها 47 وأضع صفرا في الخانات التي لم تكن ضمن عملية الجمع.

$$47 = 32 + 8 + 4 + 2 + 1 \quad \text{والعدد الثنائي الذي يساويه هو: } 101111$$

نشاط 1

1- جد العدد الثنائي المطابق للعدد العشري (132)₁₀ بالطريقة التي تناسبك.

6- تحويل عدد ما من النظام الثنائي إلى النظام العشري.

هناك طريقتان لتحويل عدد ثنائي إلى عدد عشري.

الطريقة الأولى:

مثال: لتحويل العدد $2(101111)$ إلى $10(???)$ أتبع الخطوات الآتية:

العدد الثنائي	1	1	1	1	0	1
موقع الرقم ضمن العدد الثنائي	5	4	3	2	1	0

اجمع جميع نتائج (كل رقم من الأرقام المكوّنة للعدد الثنائي مضروباً بالرقم $2^{\text{قوة موقع الرقم}}$ ضمن العدد الثنائي.

$$\begin{aligned} & 2^5 * 1 + 2^4 * 0 + 2^3 * 1 + 2^2 * 1 + 2^1 * 1 + 2^0 * 1 = \\ & 32 + 0 + 8 + 4 + 2 + 1 = \end{aligned}$$

$$47 =$$

الطريقة الثانية:

.....	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
								1	0	1	1	1	1
.....	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

أجمع الأعداد العشرية الموجودة في نفس العمود مع الأرقام (1) للعدد الثنائي.

$$47 = 1 + 2 + 4 + 8 + 16 + 32$$

1- جد العدد العشري المطابق للعدد الثنائي $(1101001)_2$ بالطريقة التي تناسبك.

2- رتب الأعداد الآتية من الأكبر إلى الأصغر.

$$_{10}(33) \mid _2(11110) \mid _{10}(44) \mid _2(10011) \mid _{10}(23) \mid _2(1101)$$

How many shortest-length paths are there to get from your house to the doughnut shop?

↑ up's
→ right's

$\binom{11}{7} = \binom{11}{4} = 330$ paths

There are six dogs to give 13 tacos. The first and sixth dog get 3 tacos, the second dog gets none, the third dog gets 5 and the fourth dog gets one.

***||☆☆☆☆|☆||☆☆☆|

$A = \{2, 4, \textcircled{1}, \textcircled{2}\}$

$(n) = \frac{n!}{(n-k)!k!}$

$e^{\pi} + 1 = 0$

P	Q	R	P ∨ Q	P ∨ R	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	T	F	F	F	F
F	F	T	F	T	F
F	F	F	F	F	F

$7, 11, 15, 19, 23, \dots$

$A_1 - A_2 = 4$
 $A_2 - A_3 = 4$
 $A_3 - A_4 = 4$
 $\vdots$
 $A_{n-1} - A_n = 4$

$3x \forall y (x \leq y + 1 \rightarrow x < 2y)$

$x - c + f = 2$

P.I.E. Example:

$4e! - \left[\binom{6}{1}5! - \binom{6}{2}4! + \binom{6}{3}3! - \binom{6}{4}2! + \binom{6}{5}1! \right]$

أهداف الدرس

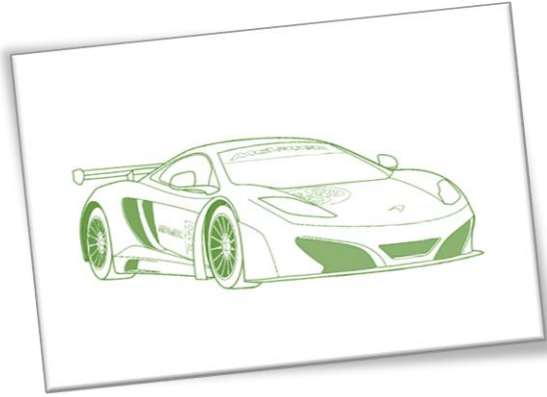


يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- تعريف الخوارزميات.
- 2- يشرح دورة حياة تطوير البرمجيات

1. تعريف الخوارزميات

نستخدم الخوارزميات في حياتنا اليومية، مثلاً إذا أردت قيادة السيارة للذهاب إلى مكان ما فإنني أتبع هذه الخطوات:



1. أفتح السيارة

2. أجلس على مقعد السائق

3. أعدّل المرآة الوسطى والمرآة الجانبية

4. أضع حزام الأمان

5. أشغل السيارة

6. أعشّق مغيّر السرعات

7. أنطلق بالسيارة

إذا فالخوارزمية تمثل تسلسلاً منطقيًا للأحداث قصد الوصول إلى الهدف أو النتيجة المطلوبة.

نشاط 1

في المثال السابق، قد يمكننا تغيير تسلسل الأحداث دون التأثير في النتيجة النهائية أو الهدف المقصود.

- غيّر بالأرقام، في تسلسل الأحداث أعلاه دون التأثير في عملية انطلاق السيارة.

--	--	--	--	--	--	--

وقد ينتج عن تغيير تسلسل الأحداث تغيير النتيجة النهائية أو استحالة التنفيذ.

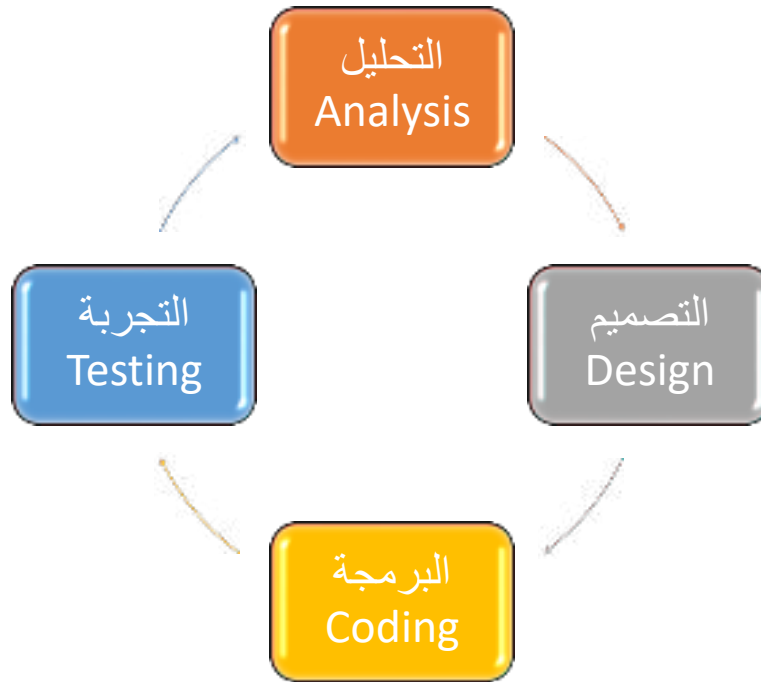
- غير بالأرقام، في تسلسل الأحداث أعلاه يحيث لا يمكن الانطلاق بالسيارة.

--	--	--	--	--	--	--

نشاط 2

أكتب في نقاط مرقمة تسلسلية منطقية مراحل طباعة الصفحة الثانية من مستند ما.

2. دورة حياة تطوير البرمجيات



المرحلة	الوصف
التحليل Analysis	في هذه المرحلة يعرف المبرمج المشكلة المراد حلّها ويحدّد الاحتياجات والمعطيات (المدخلات) المطلوبة من البرنامج وطريقة معالجتها والمخرجات المطلوبة.
التصميم Design	بعد تحديد المدخلات وطريقة معالجتها والمخرجات المنتظرة من البرمجية، تكون مرحلة التصميم لتحديد طريقة إنشاء البرمجية من خلال الخوارزميات Algorithms.
البرمجة Coding	تحويل الخوارزميات إلى لغة برمجة.
التجربة Testing	تكون بصورة مكثّفة من خلال تجربة المدخلات بكلّ أنواعها لتحديد المشاكل الناتجة عنها وتتبعها وتصحيحها بحيث تكون البرمجية خالية من أيّة شوائب وقادرة على التعامل مع المدخلات في جميع الحالات والحصول على المخرجات المطلوبة بدقّة.

اطرح مشكلة برمجية، ثم حدّد بعض الإجراءات المطلوبة في المرحلتين الأولى والثانية من مراحل حياة تطوير البرمجيات.

How many shortest-length paths are there to get from your house to the doughnut shop?

↑ + up's
→ + right's

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$

$\binom{11}{5} = \binom{11}{6} = 330$ paths

There are six dogs to give 13 tacos. The first and sixth dog get 3 tacos, the second dog gets none, the third dog gets 5 and the fourth dog gets one.

|||*||***|

$A = \{2, 4, 10, 14\}$

دروس الثالث

الكود المزيف

PSEUDOCODE

أهداف الدرس



يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- تعريف ماهية الكود المزيف
- 2- يدرك أساسيات الكتابة عن طريق الكود المزيف
- 3- يحل مسائل برمجية عن طريق الكود المزيف Pseudocode.

1. ما هو الكود المزيف Pseudocode

هي تدوين التعليمات البرمجية بلغة تشبه لغة البرمجة بطريقة مبسطة لتصميم حل لمشكلة برمجية بحيث تكون المصطلحات المستخدمة في الكود المزيف متكوّنة من الكلمات الإنجليزية المستخدمة في حياتنا اليومية. كتابة حلّ مشكلة برمجية عن طريق الكود المزيف بطريقة صحيحة تمكّنك من تحويله إلى أيّ لغة برمجة بكلّ سهولة.

مثال 1

أكتب بلغة Pseudocode برنامجاً يمكّنك من إدخال عدد صحيح وعرض رسالة تبين إن كان العدد المدخل زوجياً أو فردياً.

INPUT number

IF (number % 2 = 0) THEN / يستخدم الرمز % لإيجاد باقي عملية القسمة */

msg ← "العدد المدخل عدد زوجي"

ELSE

msg ← "العدد المدخل عدد فردي"

ENDIF

OUTPUT msg

تعليمات طريقة كتابة الكود المزيف Pseudocode

■ المسافة البادئة Indentation

تكتب التعليمات البرمجية الموجودة ضمن جملة برمجية أخرى على مسافة أبعد من هذه الأخيرة.

مثال 2

```
IF condition THEN
```

```
    ←→ Instructions
```

```
    ←→ -----
```

```
ENDIF
```

■ التعليقات comments

تحتاج بعض الأحيان إلى كتابة تعليق حول تعليمة برمجية أو معادلة لتتذكر شيئاً معيناً يخص هذه الجملة البرمجية، ولتميّز التعليق عن باقي الجمل البرمجية يجب أن يسبق التعليق الرمز: //

مثال 3

```
discount = price * (0.25) // الخصم هنا يمثل ربع السعر
```

■ الكلمات المفتاحية Keywords

هي مصطلحات لها معنى في الكود المزيف أو الحلّ الخوارزمي وتكون بأحرف انجليزية كبيرة حتى نُميّزها عن باقي العبارات الأخرى كالمتغيرات.

مثال 4

INPUT / OUTPUT / IF THEN ELSE \ WHILE ENDWHILE \ INTEGER
\ STRING \

■ أنواع البيانات Data Types

اسم النوع	الوصف	أمثلة
INTEGER	عدد صحيح	(3, -4, 20)
FLOAT, REAL	عدد عشري	(3.5, -9.6)
CHAR	حرف	("A", "a", "+", "\$")
STRING	مجموعة من الحروف المتتالية	"HELLO", "Nice To Meet You"
BOOLEAN	قيمة منطقية	(TRUE OR FALSE (1-0))
DATE	تاريخ	01/01/2000

■ تعيين المتغيرات والثوابت declaring variables/constants

المقصود بتعيين متغير/ثابت هو إسناد اسم ونوع هذا المتغير.

ملاحظة: بعض لغات البرمجة مثل Python لا تفرض ضرورة تعيين المتغيرات/الثوابت

يفضّل عند كتابة الكود المزيّف Pseudocode تعيين المتغيّرات/الثوابت كالآتي:

DECLARE VarName:DataType

مثال 5

DECLARE FirstName:STRING //

DECLARE NumberOfCars:INTEGER

DECLARE DateOfBirth:Date

CONSTANT discount=0.25

■ العوامل الحسابية mathematics operators

يبيّن الجدول ادناه العوامل الحسابية المستخدمة في كتابة الكود المزيّف Pseudocode

رمز العامل الحسابي	الوصف	مثال
+	للجمع	$x + y$
-	للطرح	$x - y$
*	للضرب	$x * y$
/	للقسمة	x / y
DIV	للحصول على الناتج الصحيح من عملية القسمة	$5 \text{ DIV } 2 = 2.5$ the result is 2
MOD	للحصول على باقي عملية القسمة	$5 \text{ MOD } 2 = 1$ OR $5 \% 2 = 1$
^	الاس أو القوة	$2^3 = 2 * 2 * 2 = 8$

■ عوامل المقارنة comparison operators

رمز عامل المقارنة	الوصف	مثال
=	يساوي	$x = y$
<>	لا يساوي	$x <> y$
<	أصغر من	$x < y$
<=	أصغر من أو يساوي	$x <= y$
>	أكبر من	$x > y$
>=	أكبر من أو يساوي	$x >= y$

■ العوامل المنطقية Logic operators

تكون نتيجة الإجراء الذي يتضمن عوامل منطقية هي **true** أو **false**.

رمز العامل المنطقي	الوصف	مثال
and	و	يجب أن يكون طرفي الجملة المنطقية صحيحان لتكون النتيجة true <u>$5 > 2$</u> and <u>$5 < 10$</u> : the result is true <u>$3 > 1$</u> and <u>$3 > 5$</u> : the result is false <u>$4 < 2$</u> and <u>$4 <= 4$</u> : the result is true
or	أو	يكفي أن يكون أحد طرفي الجملة المنطقية صحيحا لتكون نتيجة الجملة المنطقية true <u>$6 > 3$</u> or <u>$6 > 10$</u> : the result is true <u>$3 < 1$</u> or <u>$3 > 5$</u> : the result is false
not	لا	not($3 > 1$): the result is false not (<u>$3 > 1$</u>) or <u>$3 > 4$</u> : the result is false

مثال 6

جد نتيجة الجمل المنطقية الآتية، مع العلم أنّ $x=4$ و $y=2$ و $z=6$:

$x > y$ or $z > y$:

$\text{not}(x=y)$ and $z < y$:

$y <> z$ and $\text{not}(x=y \text{ or } x > (z-y))$:.....

■ الاسناد assignment

هو إجراء يتمثل في اسناد قيمة لمتغير ما.

مثال 7

$x \leftarrow 4$

$y \leftarrow 2.3$

$\text{name} \leftarrow \text{"ahmad"}$

$\text{Total} \leftarrow x + y$

نشاط 2

Variables	Data Types
power	integer
speed	Reel/float/decimal
statement	string



Sale	date
------	------

أسند كل قيمة من القيم الموجودة في الدائرة إلى المتغير المناسب.

■ جملة ادخال البيانات INPUT statement

تنفيذ هذه الجملة البرمجية يجعل الحاسوب ينتظر المستخدم لإدخال قيمة معينة من أحد أجهزة الإدخال والتي في معظم الأحيان تكون لوحة المفاتيح.

مثال 8

ينتظر الحاسوب المستخدم لإدخال درجة طالب مثلا // **INPUT degree**

لإدخال الاسم الأول لشخص ما // **INPUT FirstName**

لإدخال إجابة عن سؤال معين // **INPUT Answer**

■ جملة عرض المعلومات OUTPUT statement

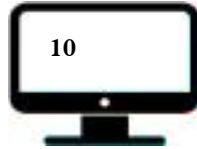
بعد إدخال البيانات أو المعطيات المطلوبة إلى جهاز الحاسوب، يقوم هذا الأخير بمعالجتها بناء على تعليمات برمجية محددة للحصول على النتائج المرجوة والتي تريد غالبا أن يتم عرضها على شاشة الحاسوب وهو المر الذي نحصل عليه عند استخدامنا جملة العرض أو جملة المخرجات.

مثال 9

OUTPUT "Welcome"



OUTPUT 5*2



INPUT Name // المستخدم أحمد يدخل اسمه

OUTPUT Name



أكتب كوداً مزيّفاً Pseudocode يجعل الحاسوب يستقبل قيمتان صحيحتان ويحفظ نتيجة مجموعهما في متغيّر باسم SumVar ويعرض النتيجة على الشاشة.

■ الجملة الشرطيّة IF – THEN – ELSE statement

مثال: أرغب في حلّ مسألة بلغة Pseudocode، لبرنامج يمكّنني من إدخال المعدّل الفصلي للطالب وإظهار الرسالة "مبارك عليك النجاح" أو "تستطيع النجاح إذا بذلت مجهوداً أكبر" بناءً على درجة النجاح التي تساوي 60 فما فوق.

الحلّ بلغة Pseudocode:

INPUT ("Enter your degree:", degree)

IF degree >= 60 **THEN**

OUTPUT ("مبارك عليك النجاح")

ELSE

OUTPUT ("تستطيع النجاح إذا بذلت مجهوداً أكبر.")

ENDIF

أكتب بلغة Pseudocode برنامجاً يمكنك من حساب سعر تذكرة الدخول إلى حلبة سباق البحرين للفورمولا 1 بناءً على عمر الشخص بحيث يتمتع الشخص الذي يبلغ من العمر 18 سنة أو أقلّ بتخفيض 25%.

المعطيات:

○ سعر التذكرة 35 دب

▪ الجمل التكرارية / Loop statements

○ جملة WHILE

WHILE condition DO

instructions

ENDWHILE

تمكّني هذه الجملة التكرارية من تكرار التعليمات **instructions**
عدد غير محدّد من المرات إلى حين تحقّق الشرط **Condition**

مثال: أرغب في حلّ مسألة تتمثّل في إدخال عدد فرديّ وأن يكرّر الحاسوب طلب إدخال هذا العدد إذا كان زوجياً.

الحلّ بلغة Pseudocode:

INPUT ("Enter an odd number:",nb)

WHILE nb MOD 2 = 0 **DO**

INPUT ("Enter an odd number:",nb)

ENDWHILE

أكتب بلغة Pseudocode برنامجاً يمكنك من إعادة طلب إدخال عددين صحيحين (قيمتهم الأولى 1 لكل منهما) حتى تصبح نتيجة ضربهما أكبر أو تساوي 1000.

○ جملة FOR

FOR i=start TO end DO

instructions

ENDFOR

تمكّني هذه الجملة التكرارية من تكرار التعليمات **instructions** عدد محدّد من المرات حيث يبدأ العدّاد i من القيمة **start** وحتى القيمة **end** وهي عدد مرات التكرار.

مثال: أرغب في حلّ مسألة تتمثّل في حساب مجموع الأعداد من 15 إلى 20.

الحلّ بلغة Pseudocode:

sum=0

FOR i=15 to 20 DO

sum = sum + i

ENDFOR

أكتب بلغة Pseudocode برنامجاً يمكنك من حساب وعرض مضاعفات العدد 8 إلى حدود 5 مرات.

How many shortest-length paths are there to get from your house to the doughnut shop?

$(n) = \frac{n!}{(n-k)!k!}$

$\binom{11}{2} = \binom{11}{+} = 330$ paths

There are six dogs to give 13...

use a "stars and bars" diagram to illustrate the...

and sixth dog get 3 tacos, the second dog gets none, the third dog gets 5 and the fourth dog gets one.

☆☆☆||☆☆☆☆☆|☆||☆☆☆|

$A = \{2, 4, \textcircled{1}, \textcircled{2}\}$

$\exists x \forall y (x \leq y + 1 \rightarrow x < 2y)$

$v = e + f = 2$

P.I.E. Example:

$4e! = \left[\binom{6}{1} 5! - \binom{6}{2} 4! + \binom{6}{3} 3! - \binom{6}{4} 2! + \binom{6}{5} 1! \right]$

الدرس الرابع

الخرائط التدفقية

FLOWCHARTS

أهداف الدرس



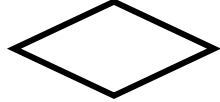
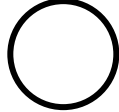
يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- يتعرف مفهوم ووظيفة الخرائط التدفقية.
- 2- التعرف على أشكال مكونات الخريطة التدفقية.
- 3- توظيف أشكال الخريطة التدفقية في المكان المناسب.
- 4- رسم الخرائط التدفقية Flowcharts لحل مشكل برمجي.
- 5- توظيف برمجية visio لرسم الخرائط التدفقية.

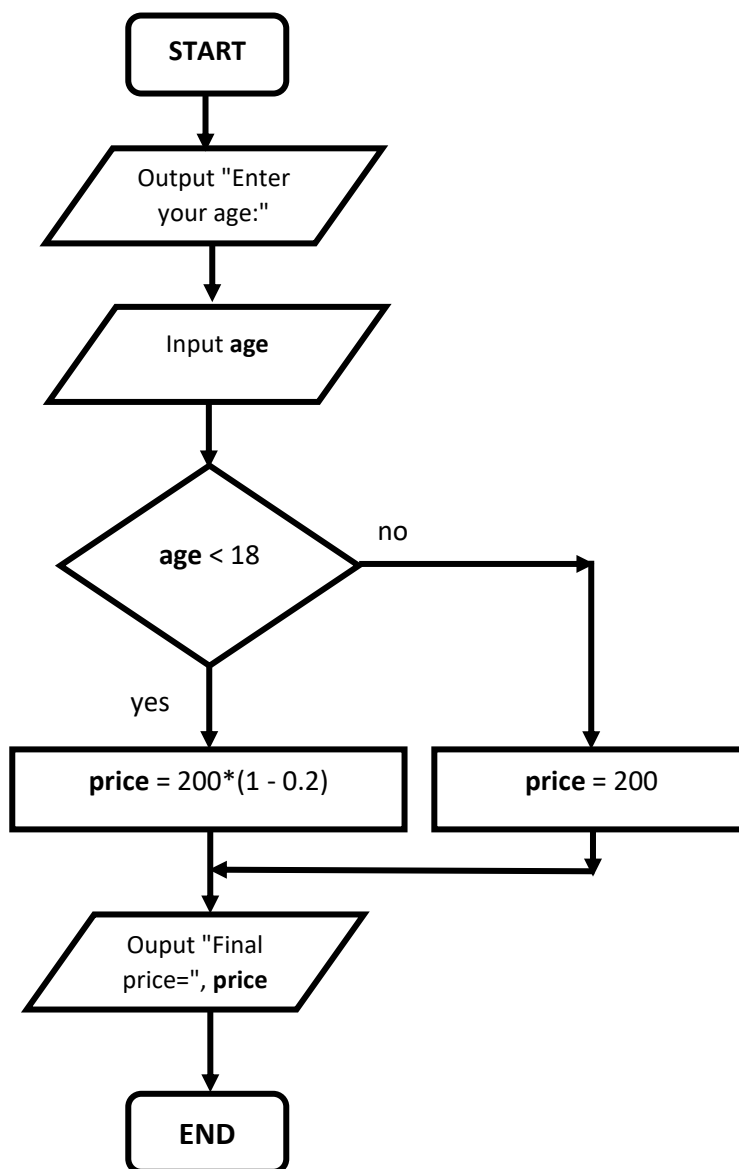
1. الخرائط التدفقية

الخريطة التدفقية هو حلّ رسوميّ للمشكلة البرمجية، حيث ترتبط مجموعة من الأشكال الهندسية بعضها ببعض في ترتيب منطقي لتسلسل الأحداث والإجراءات البرمجية للحلّ الخوارزمي، ويحتوي كلّ شكل من الأشكال على جمل خوارزمية تبين الاجراء المتّبع في هذه النقطة من البرنامج.

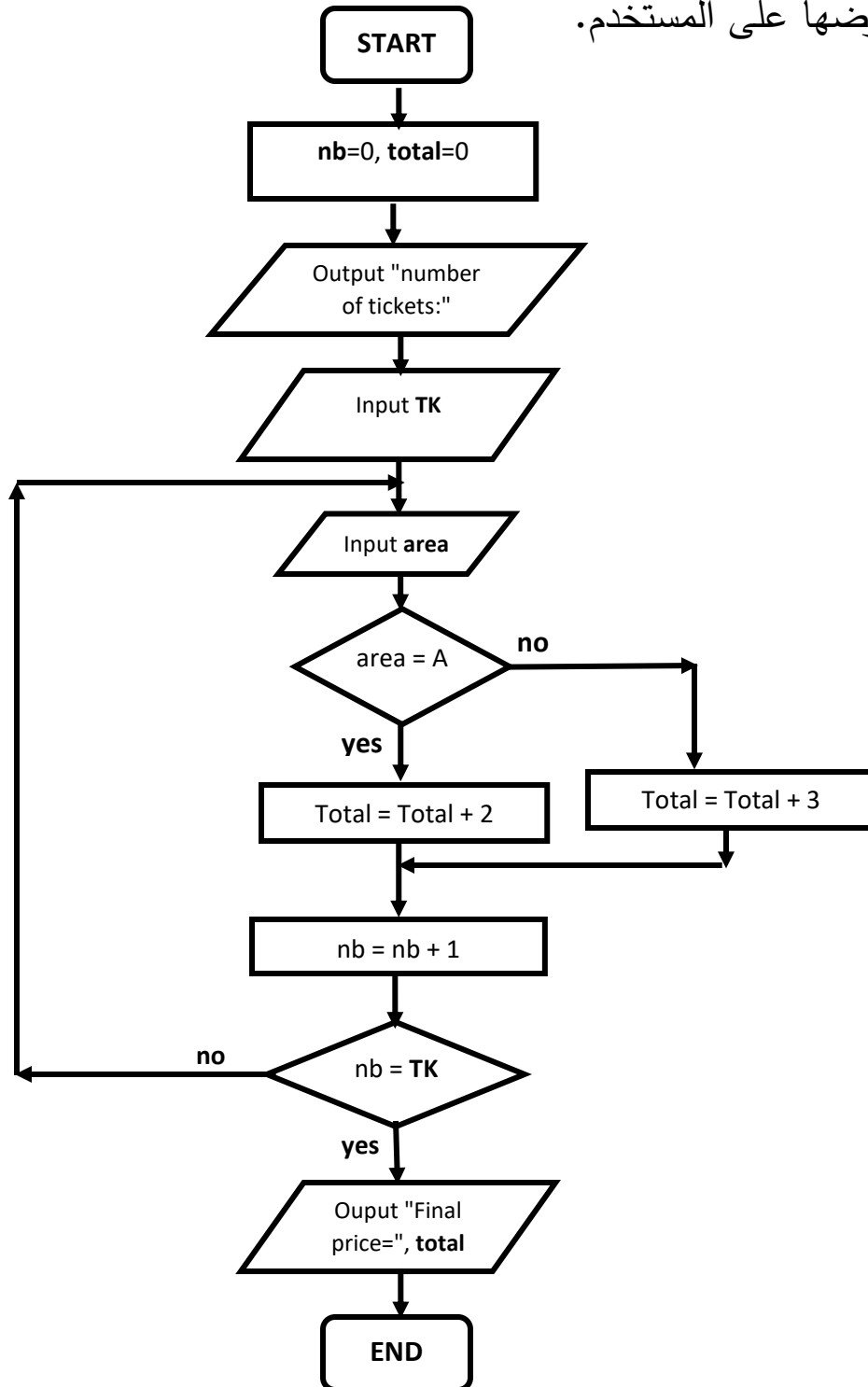
الأشكال المكوّنة للخريطة التدفقية:

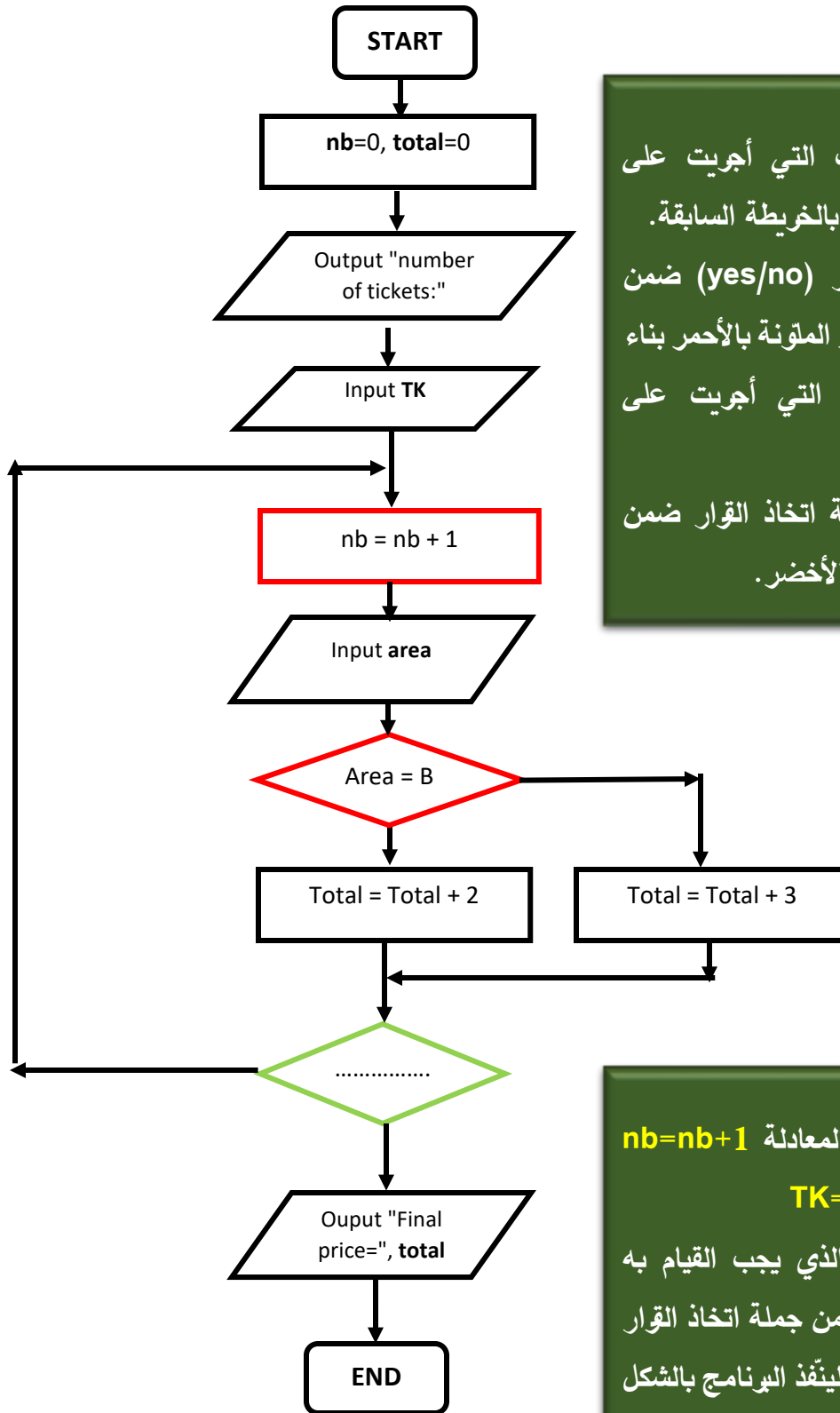
الشكل	الاسم: عربي/English	الوصف
	البداية/النهاية Start/End	يستخدم في بداية ونهاية الخريطة التدفقية
	مدخلات/مخرجات Input/output	يستخدم عند إدخال المعطيات و/أو عرض المخرجات
	معالجة process	عمليات حسابية منطقية، تعليمة برمجية
	اتخاذ القرار Decision	عندما يكون هناك اجراء سيتّخذ بناء على شرط نتيجته (نعم/لا)
	الاتجاه Flow arrow	يبيّن اتجاه الارتباط بين مختلف أشكال الخريطة التدفقية
	الرابط connector	للقفز من نقطة محدّدة من البرنامج إلى نقطة أخرى
	مقطع برمجي Subroutine	مقطع برمجي فرعي في موقع ما من البرنامج

يبلغ سعر تذكرة طائرة إلى دولة ما 200 د.ب، فإذا كان سنّ المسافر أقلّ من 18 سنة فإنّه سيتمّ بـ 20% خصم. أرسم الخريطة التدفقية لحساب السعر النهائي للتذكرة وعرضها للمستخدم.



اتّفق مجموعة من الأصدقاء على دخول السينما، وكان سعر التذكرة 2 دب للشخص في المنطقة **A** و 3 دب للشخص في المنطقة **B**. أرسم الخريطة التدفقية لحساب السعر الإجمالي للتذاكر وعرضها على المستخدم.





3. وضح التعديلات التي أجريت على البرنامج مقارنة بالخريطة السابقة.
4. قم بتعديل القوار (yes/no) ضمن جمل اتخاذ القوار الملونة بالأحمر بناء على التعديلات التي أجريت على البرنامج.
5. أكمل شرط جملة اتخاذ القوار ضمن الشكل الملون بالأخضر.

1. لو قمت بتعديل المعادلة $nb=nb+1$ بالتالي: $TK=TK-1$
2. ما هو التعديل الذي يجب القيام به على البرنامج ضمن جملة اتخاذ القوار باللون الأخضر، لينفذ البرنامج بالشكل الصحيح.

رسوم الفحص الفني للسيارات = 20دب، إذا علمت أن السيارات التي يتجاوز عمرها 10 سنوات تضاف إلى تكلفة فحصها 10% والتي يتجاوز عمرها 15 سنة يضاف إلى رسوم الفحص 20%.

أكتب في خطوات طريقة حساب مبلغ رسوم الفحص النهائي.

أرسم الخريطة التدفقية لحساب قيمة رسوم الفحص وعرضه على المستخدم.

How many shortest-length paths are there to get from your house to the doughnut shop?

↑ up's
→ right's

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$

$\binom{11}{5} = \binom{11}{6} = 330$ paths

There are six dogs to give 13 tacos. The first and sixth dog get 3 tacos; the second dog gets none; the third dog gets 5 and the fourth dog gets one.

***||☆☆☆☆|☆||☆☆☆|

$A = \{2, 4, \textcircled{1}, \textcircled{2}\}$

$(n) = \frac{n!}{(n-k)!k!}$

P	Q	R	P ∨ Q	P ∨ R	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	T	F	F	F	F
F	F	T	F	T	F
F	F	F	F	F	F

$7, 11, 15, 19, 23, \dots$

$x_1 - x_2 = 4$
 $x_2 - x_3 = 4$
 $x_3 - x_4 = 4$
 $\vdots$
 $x_{n-1} - x_n = 4$

$3x \forall y (x \leq y + 1 \rightarrow x < 2y)$

$v = c + f = 2$

P.I.E. Example:

$4! - \left[\binom{6}{1}5! - \binom{6}{2}4! + \binom{6}{3}3! - \binom{6}{4}2! + \binom{6}{5}1! \right]$

أهداف الدرس



يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- التعرف على أنواع البيانات المختلفة التي سيتعامل معها برمجياً.
- 2- يميز بين أنواع البيانات المختلفة.

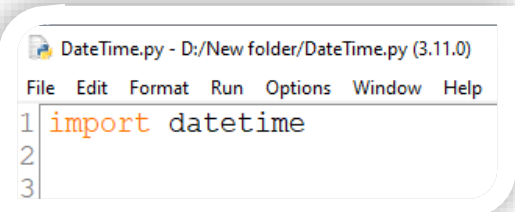
■ أنواع البيانات Data Types

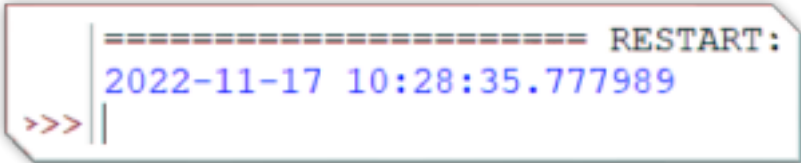
PSEUDOCODE

أمثلة	الوصف	اسم النوع
(3, -4, 20)	عدد صحيح	INTEGER
(3.5, -9.6)	عدد عشري	FLOAT, REAL
("A", "a", "+", "\$")	حرف	CHAR
"HELLO", "Nice To Meet You"	مجموعة من الحروف المتتالية	STRING
[10,5,8,6,4] ['a','A','b','f'] ['hello','world','ab100']	مصفوفة	array
(TRUE OR FALSE (1-0))	قيمة منطقية	BOOLEAN
01/01/2000	تاريخ	DATE

■ أنواع البيانات Data Types

PYTHON

اسم النوع	الوصف	أمثلة
integer	عدد صحيح	(3, -4, 20)
float	عدد عشري	(3.5, -9.6)
string	حرف/حروف	"A", "a", "+", "\$", "Welcome to Bahrain"
list	مصفوفة	x=["mouse","keyboard","memory"]
bool	قيمة منطقية	b= false
خصوصية بعض أنواع البيانات في لغة Python		
date	لا يوجد	يتعامل البايثون مع المتغيرات التي تحفظ تاريخاً من خلال استدعاء سكريبت برمجيّ مضمّن يسمّى "datetime"
		للتمكن من التعامل مع بيانات من نوع تاريخ date في لغة Python يجب استدعاء المكتبة الخاصة بالتاريخ في بداية المقطع البرمجي كالآتي:
		

<pre>import datetime print(datetime.datetime.now())</pre> النتيجة: 	لعرض التاريخ والوقت الحالي	
<pre>import datetime print(datetime.datetime.now().year)</pre> مثال التاريخ الحالي هو: 2022-11-20 النتيجة: 2022	لعرض السنة فقط من التاريخ الحالي	date
<pre>import datetime print(datetime.datetime.now().month)</pre> مثال مثال التاريخ الحالي هو: 2022-11-20 النتيجة: 11	لعرض الشهر فقط من التاريخ الحالي	
<pre>import datetime print(datetime.datetime.now().day)</pre> مثال التاريخ الحالي هو: 2022-11-20 النتيجة: 20	لعرض اليوم فقط من التاريخ الحالي	
<pre>import datetime print(x.strftime("%A"))</pre> مثال: التاريخ 2022-11-20 النتيجة: Sunday	لعرض اسم اليوم فقط من التاريخ	

<pre>import datetime print(x.strftime("%B"))</pre> مثال: التاريخ 20-11-2022 النتيجة: November	لعرض اسم الشهر فقط من التاريخ																													
<pre>var=datetime.datetime(year, month, day)</pre> مثال: <pre>import datetime d=datetime.datetime(2022, 12, 31)</pre> لعرض التاريخ: <code>print(d)</code> → 2022-12-31 00:00:00 لعرض السنة: <code>print(d.year)</code> → 2022 لعرض الشهر: <code>print(d.month)</code> → 12 لعرض اليوم: <code>print(d.day)</code> → 31	لإسناد تاريخ إلى متغير																													
مثال1: أرغب في حفظ الأرقام الفردية من 1 إلى 9 في متغير واحد. فيكون شكل المتغير عند اسناد قيم له كالآتي: <pre>unp=[1,3,5,7,9]</pre> مثال2: أرغب في حفظ حروف وكلمات في متغير واحد، فيكون شكل المتغير عند اسناد قيم له كالآتي: <pre>charword=["a","abc","welcome","Bahrain","e"]</pre> <u>الوصول إلى قيمة محدّدة ضمن المصفوفة:</u> <table><tr><td>المغیر</td><td>values</td><td>1</td><td>3</td><td>5</td><td>7</td><td>9</td></tr><tr><td>unp</td><td>index</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> <table><tr><td>المغیر</td><td>values</td><td>"a"</td><td>"abc"</td><td>"welcome"</td><td>"Bahrain"</td><td>"e"</td></tr><tr><td>charword</td><td>index</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	المغیر	values	1	3	5	7	9	unp	index	0	1	2	3	4	المغیر	values	"a"	"abc"	"welcome"	"Bahrain"	"e"	charword	index	0	1	2	3	4	- لحفظ مجموعة من القيم في متغير واحد - تسمح بتكرار القيم - قابل للتغيير	تسلسلي list
المغیر	values	1	3	5	7	9																								
unp	index	0	1	2	3	4																								
المغیر	values	"a"	"abc"	"welcome"	"Bahrain"	"e"																								
charword	index	0	1	2	3	4																								

ألاحظ أنّ لكلّ قيمة في المصفوفة، موقع محدّد يسمّى في لغة البرمجة **index**، كما هو مبين في تصميم المصفوفة أعلاه.

يمكنني الـ **index** من تحديد محتوى المتغيّر في هذا الموقع بالشكل الآتي:

`print(unp[index])` → 

مثال:

`print(unp[0])` → 

`print(unp[3])` → 

`print(charword[4])` → 

`print(charword[1])` → 

مثال:

```
d={ 'mob':'samsung',
    'ram':'4GB',
    'storage':'256GB',
    'Qty':20
}
```

`print(d)→`

{'mob':'samsung','ram':'4GB','storage':'256GB','Qty':20}

`print(d['mob'])` → samsung

`print(d['ram'])` → 4GB

- لحفظ
مجموعة
من
العناصر
في متغيّر
واحد
- كل عنصر
يتكون من
مفتاح
القيمة
والقيمة.

تسلسلي
dict

<pre>print(d['storage']) ➔ 256GB print(d['Qty']) ➔ 20</pre>	- لا تسمح بتكرار العناصر - قابل للتغيير	
---	---	--

How many shortest-length paths are there to get from your house to the doughnut shop?

↑ = up
→ = right

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$

$\binom{11}{5} = \binom{11}{6} = 330$ paths

There are six dogs to give 13 tacos. Use a "stars and bars" diagram to illustrate the problem. The first and sixth dog get 3 tacos; the second dog gets none; the third dog gets 5 and the fourth dog gets one.

***||☆☆☆☆|☆||☆☆☆|

$A = \{2, 4, \textcircled{1}, \textcircled{2}\}$

$(n) = \frac{n!}{(n-k)!k!}$

P	Q	R	P ∨ Q	P ∨ R	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	T	F	F	F	F
F	F	T	F	T	F
F	F	F	F	F	F

$7, 11, 15, 19, 23, \dots$

$A_1 - A_2 = 4$
 $A_2 - A_3 = 4$
 $A_3 - A_4 = 4$
 $\vdots$
 $A_{n-1} - A_n = 4$

$3x \forall y (x \leq y + 1 \rightarrow x < 2y)$

$x - c + f = 2$

P.I.E. Example:

$4! = \left[\binom{6}{1}5! - \binom{6}{2}4! + \binom{6}{3}3! - \binom{6}{4}2! + \binom{6}{5}1! \right]$

أهداف الدرس



يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- تعيين المتغيرات والثوابت بالطريقة الصحيحة ضمن الكود المزيف
- 2- يميز أنواع البيانات حسب محتوى المتغير.

■ قواعد تسمية المتغيرات والثوابت

PSEUDOCODE / PYTHON

- يجب أن تتكوّن فقط من حروف وأرقام. (A-Z,a-z,0-9)
- لا يجب أن يبدأ برقم.
- يمكن أن يكون مزيجاً من الحروف الكبيرة والصغيرة (مثال: TotalToPay)
- لا تستخدم الكلمات المفتاحية للغة البرمجة كأسماء متغيرات أو ثوابت.

نشاط 1

حدّد في الجدول الآتي، ما إذا كانت تسميات المتغيرات/الثوابت صحيحة أو لا! ولماذا؟

variable/Constant	Correct	Incorrect	If incorrect: Why?
12Amount			
T3b2			
Integer			
Size1			
Am%nt			

■ تعيين المتغيرات والثوابت declaring variables/constants

PSEUDOCODE

المقصود بتعيين متغير/ثابت هو إسناد اسم ونوع هذا المتغير.

يفضّل عند كتابة الكود المزيّف Pseudocode تعيين المتغيرات/الثوابت كالآتي:

DECLARE VarName:DataType

```
DECLARE FirstName:STRING //  
DECLARE NumberOfCars:INTEGER  
DECLARE DateOfBirth:Date  
CONSTANT discount=0.25
```

PYTHON

لا تفرض لغة **PYTHON** تعيين المتغيرات ونوعها قبل استخدامها. بإمكانني استخدام أيّ متغير مباشرة، بحيث يكون نوع المتغير هو نفس نوع القيمة التي أسندت إليه.

Arithmetic & Boolean & Comparison operators

2- توظيف العامل المناسب في المكان المناسب من الجملة اليرمجية.

العوامل الحسابية mathematics operators

PSEUDOCODE

يبيّن الجدول ادناه العوامل الحسابية المستخدمة في كتابة الكود المزيّف Pseudocode

رمز العامل الحسابي	الوصف	مثال
+	للجمع	$x + y$
-	للطرح	$x - y$
*	للضرب	$x * y$
/	للقسمة	x / y
DIV	للحصول على الناتج الصحيح من عملية القسمة	$5 \text{ DIV } 2 = 2.5$ the result is 2
MOD	للحصول على باقي عملية القسمة	$5 \text{ MOD } 2 = 1$ OR $5 \% 2 = 1$
^	الاس أو القوة	$2^3 = 2 * 2 * 2 = 8$

PYTHON

رمز العامل الحسابي	الوصف	مثال
+	للجمع في حال البيانات الرقمية	If $x=2$ & $y=3 \rightarrow x + y=5$
+	لربط في حال البيانات النصية	If $x="ali"$ & $y="fahd" \rightarrow x+y=alifahd$
-	للطرح	$x - y$
*	للضرب	$x * y$
/	للقسمة	x / y
//	للحصول على الناتج الصحيح من عملية القسمة	$5 // 2 = 2.5$ the result is 2
%	للحصول على باقي عملية القسمة	$5 \% 2 = 1$
**	للحصول على قوة العدد	$5 ** 2 = 25$

عوامل المقارنة comparison operators

PSEUDOCODE

رمز عامل المقارنة	الوصف	مثال
=	يساوي	$x = y$
<>	لا يساوي	$x <> y$
<	أصغر من	$x < y$
<=	أصغر من أو يساوي	$x <= y$
>	أكبر من	$x > y$
>=	أكبر من أو يساوي	$x >= y$

PYTHON

رمز عامل المقارنة	الوصف	مثال
==	يساوي	$x == y$
!=	لا يساوي	$x != y$
<	أصغر من	$x < y$
<=	أصغر من أو يساوي	$x <= y$
>	أكبر من	$x > y$
>=	أكبر من أو يساوي	$x >= y$

العوامل المنطقية Logic operators

PYTHON & PSEUDOCODE

تكون نتيجة الإجراء الذي يتضمن عوامل منطقية هي true أو false.

رمز العامل المنطقي	الوصف	مثال
and	و	يجب أن يكون طرفي الجملة المنطقية صحيحان لتكون النتيجة true: <u>5>2</u> and <u>5<10</u> : the result is true <u>3>1</u> and <u>3>5</u>: the result is false <u>4<>2</u> and <u>4<=4</u>: the result is true
or	أو	يكفي أن يكون أحد طرفي الجملة المنطقية صحيحا لتكون نتيجة الجملة المنطقية true: <u>6>3</u> or <u>6>10</u> : the result is true <u>3<1</u> or <u>3>5</u>: the result is false
not	لا	not(3>1): the result is false not(<u>3>1</u>) or <u>3>4</u>: the result is false

نشاط 1

جُد نتيجة الجُمْل المنطقية الآتية، مع العلم أنَّ $x=4$ و $y=2$ و $z=6$:

$x > y$ or $z > y$:

$\text{not}(x=y)$ and $z < y$:

$y < > z$ and $\text{not}(x=y \text{ or } x > (z-y))$:

نشاط 2

جُد نتيجة الجُمْل الحسابية الآتية في لغة Python:

$13/2$:

$13 \% 2$:

$13 // 2$:

$10 ** 2$:

■ جملة ادخال البيانات INPUT statement

PSEUDOCODE

تنفيذ هذه الجملة البرمجية يجعل الحاسوب ينتظر المستخدم لإدخال قيمة معينة من أحد أجهزة الإدخال والتي في معظم الأحيان تكون لوحة المفاتيح.

مثال 1

INPUT *degree* // *degree* إدخال درجة طالب مثلا وحفظها في المتغير *degree*

INPUT *FirstName* // *FirstName* لإدخال الاسم الأول لشخص ما وحفظ المدخل في المتغير *FirstName*

INPUT *Answer* // *Answer* إدخال إجابة عن سؤال معين

إدخال بيانات مرفقة برسالة

INPUT ("Plz Enter your answer:", *Answer*)

PYTHON

في لغة Python لإدخال بيانات وحفظها في متغير، نستخدم البناء التالي:

variable = **input**("message:")

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48)
[MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> degree=input("Enter the student degree:")
```

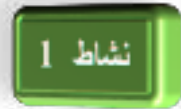
جملة إدخال بيانات لحفظها في المتغير *degree*

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48)
[MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> degree=input("Enter the student degree:")
Enter the student degree:
```

بعد الضغط على المفتاح **Enter** يتم تنفيذ الجملة البرمجية، ويُنتظر مَنِي إدخال البيانات

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48)
[MSC v.1933 64 bit (AMD64)] on win7
Type "help", "copyright", "credits()" or "license()" for more
>>> degree=input("Enter the student degree:")
Enter the student degree:91
>>>
```

أقوم بإدخال القيمة 91 مثلاً وأضغط على المفتاح Enter فيتم حفظ القيمة المدخلة في المتغير **degree**



ضمن الملف السابق، أكمل إدخال بقية الأوامر البرمجية لإدخال الاسم الأول ضمن المتغير **FirstName**.

هام جداً

الدالة int() : لتحويل قيمة نصية تتكوّن من أرقام إلى قيمة رقمية.

عند إدخال بيانات رقمية عن طريق الجملة البرمجية الآتية:

لنفترض ادخال الرقم 7 // `Num = input("Enter a number")`

يكون نوع المتغير **Num** نصياً بالرغم من أنّ البيانات المدخلة رقمية يعني **Num** تحوي الرمز (7) وليس الرقم 7.

إذا أردت بعد ذلك استخدام المتغير **Num** في معادلة حسابية، مثلاً كالاتي: `Total=Num+2`

فسوف ينتج عن ذلك خطأ برمجياً لأنه لا يمكن جمع بيانات نصية مع بيانات رقمية.

تتم معالجة هذا الخطأ من خلال تحويل قيمة المتغير **Num** من نصية إلى رقمية باعتماد الدالة **int()** فتصبح المعادلة كالاتي:

`Total = int(Num) + 2 // result is 9`

الدالة str() : لتحويل قيمة رقمية إلى قيمة نصية.

استنتج حل المعادلة الآتية: `Total=Num + str(2)`

■ جملة عرض المعلومات OUTPUT statement

PSEUDOCODE

بعد إدخال البيانات أو المعطيات المطلوبة إلى جهاز الحاسوب، يقوم هذا الأخير بمعالجتها بناء على تعليمات برمجية محدّدة للحصول على النتائج المرجّوة والتي نريد غالباً أن يتمّ عرضها على شاشة الحاسوب وهو الأمر الذي نحصل عليه عند استخدامنا جملة العرض أو جملة المخرجات:

OUTPUT *value || variable || equation*

مثال 2

OUTPUT "Welcome"

OUTPUT 5*2

INPUT Name // المستخدم أحمد يدخل اسمه

OUTPUT Name



ناقش مع معلّمك وأصدقائك، متى يتمّ استخدام علامات التنصيص في جملة الادخال والإخراج.

PYTHON

في لغة Python لعرض مخرجات ما على شاشة الحاسوب، أتّبع البناء التالي:

print(*value || variable || equation*)

نشاط 2

في المثال السابق إذا أردت التأكّد من محتوى المتغيّر **degree** بعد إدخال درجة الطالب أقوم بما هو مبين في الصور أسفله.

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48)
[MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> degree=input("Enter the student degree:")
Enter the student degree:
>>> print(degree)
```

أدخل جملة العوض: `print(degree)`

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48)
[MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> degree=input("Enter the student degree:")
Enter the student degree:91
>>> print(degree)
91
>>>
```

أضغط على المفتاح `Enter` وألاحظ عرض قيمة المتغير `degree`

■ الاسناد assignment

هو إجراء يتمثل في اسناد قيمة لمتغير ما.

مثال 3

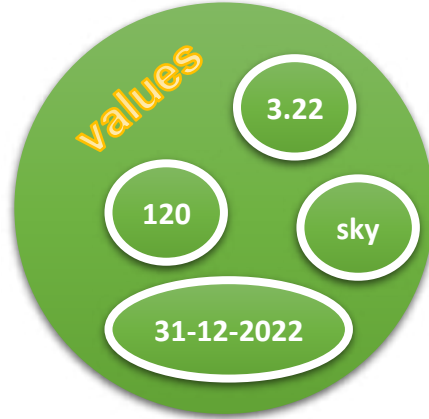
`x ← 4`

`y ← 2.3`

`name ← "ahmad"`

`Total ← x + y`

Variables	Data Types
power	integer
speed	Reel/float/decimal
statement	string
Sale	date



1. أسند كل قيمة من القيم الموجودة في الدائرة إلى المتغير المناسب من الجدول.

2. افتح محرر لغة البرمجة Python

3. احفظ الملف في مجلدك الخاص باسم **assign.py**

4. قم بتطبيق ما نفذته في السؤال الأول.

5. اعرض محتوى المتغيرات.

1. أكتب كوداً مزيقاً **Pseudocode** ينفذ الآتي:

1. يعين متغيران صحيحان

2. يجعل الحاسوب يستقبل قيمتان صحيحتان يحفظان في هذين المتغيرين.

3. يحفظ نتيجة مجموعهما في متغير باسم **SumVar** ويعرض النتيجة على الشاشة.

4. يحفظ باقي قسمة **SumVar** على العدد 3 في المتغير **Multi3**.

١١. نفّذ الآتي:

1. افتح ملف **Python** جديد واحفظه باسم **InputOutput.py**
2. ضمن الملف الجديد، حوّل الكود المزيّف إلى لغة **Python**.
3. نظريًا: هل العدد **SumVar** من مكرّرات الرقم 3؟

نشاط 5

أكتب كودًا مزيّفًا **Pseudocode** ينفّذ الآتي:

1. يعيّن سلسلة من 5 أعداد صحيحة.
2. يعكس قيمة العنصر الأول مع العنصر الأخير من السلسلة
3. يظهر عناصر السلسلة بعد التبديل.
4. أرسم الخريطة التدفقية للحل بالاستعانة ببرمجيّة **visio**.
5. حوّل الكود إلى لغة **Python**.

الدرس التاسع

الدوال المضمّنة

BUILT IN FUNCTIONS

<https://docs.python.org/3/library/functions.html>

أهداف الدرس

يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- التعرف على وظائف أهم الدوال المضمّنة في لغة Python.
- 2- توظيف الدالة المناسبة في الموقع المناسب لحلّ مشكل برمجي.

■ ما هي الدوال؟

الدالة هي كتلة من الجمل البرمجية الجاهزة مسبقاً، يتم استدعاؤها عند اللزوم، عادة ما ترافقها بيانات خاصّة بها تسمى "parameters" لاستخدامها للوصول إلى النتيجة المطلوبة. عادة ما تقوم الدالة بإرجاع النتيجة إلى البرنامج الرئيسي الذي تم استدعاؤها منه.

مثال

1- الدالة `abs()` وهي دالة القيمة المطلقة وتستخدم كالآتي:

$$X = -5$$

$$Y = \text{abs}(X) \rightarrow Y=5$$

2- الدالة `len()` وهي دالة لإرجاع عدد عناصر مصفوفة ما:

`Fruits = {'banana', 'apple', 'orange', 'pineapple'}`

$$\text{Nb} = \text{len}(\text{Fruits}) \rightarrow \text{Nb} = 4$$

■ الدوال المطلوبة لاستخدامها مع البيانات الرقمية والنصية

اسم الدالة	الوظيفة	مثال
int()	لتحويل قيمة نصية إلى قيمة رقمية	x="5" y = int(x) → y=5
str()	لتحويل قيمة رقمية إلى قيمة نصية	x=5 y = str(x) → y=5 (الـ 5 هنا كرمز وليس كرقم)
range()	لتحديد مجال رقمي	x = range(6) بمعنى أنّ المتغيّر x يحتوي الأرقام من 0 إلى 5 البنية العامة للدالة range(): range(start,stop,step) start: رقم بدء المجال (افتراضيا 0) stop: رقم نهاية المجال (بحيث لا ينتمي هذا الرقم إلى المجال المحدد) step: خطوة الزيادة. مثال: a = range(1,9,2) for b in a: print(b) فتكون النتيجة كالآتي: 1 3 5 7
abs()	لإرجاع القيمة المطلقة لعدد معين	x = -5 y = abs(x) → y = 5
float()	لتحويل عدد صحيح إلى عدد عشري	x = 2 y = float(x) → y = 2.0
len()	لإرجاع عدد عناصر مصفوفة أو قيمة نصية	x= {1,5,10} y = len(x) → y = 3 x='ahmed' y = len(x) → y = 5

	للجبر إلى الأقرب	round()
x=[0,2,5,-1] m=min(x) print(m) Result: -1	للبحث عن أصغر قيمة ضمن سلسلة	min()
x=[0,2,5,-1] mx=max(x) print(mx) Result: 5	للبحث عن أكبر قيمة ضمن سلسلة	max()
x=[0,2,5,-1] s=sum(x) print(s) Result: 6	لجمع أعداد سلسلة	sum()
x=[0,2,5,-1] srt=sorted(x) print(srt) Result: [-1,0,2,5]	لترتيب تصاعديا سلسلة	sorted()
x=[0,2,5,-1] srt=sorted(x, reverse=true) print(srt) Result: [5,2,0,-1]	لترتيب سلسلة تنازليا	sorted(list,reverse=true)
cities=['manama', 'muharraq', 'hid'] ind = cities.index(hid) Result is: ind = 2	لتحديد موقع عنصر ضمن سلسلة	index()
cars=[Math, 'Science', 'English'] cars.remove('Science') print(cars) Result is: ['Math', 'English']	لحذف عنصر من سلسلة بناء على قيمه.	remove()

<pre># random ضرورة استدعاء الوحدة import random x= random.random() print(x) Result may be: 0.325415....</pre>	1- لاختيار عدد عشري عشوائي بين 0.1 و 1.0	
<pre>import random x= random.randint(10, 20) print(x) Result may be: 13</pre>	2- لاختيار عدد عشوائي صحيح. (مثلا بين 10 و 20)	
<pre>import random x= random.randrange(15) print(x) Result may be: 9</pre>	3- لاختيار عدد عشوائي صحيح ضمن مجال محدّد بدءاً من الصفر. مثال: (بين 0 و 15: 15 غير محتسبة من ضمن المجال)	random لاختيار عدد عشوائي، ولها عدّة دوال مصاحبة سنعدّد أهمّها تباعاً:
<pre>import random x= random.randrange(5, 30) print(x) Result may be: 29</pre>	4- لاختيار عدد عشوائي صحيح ضمن مجال تحدّد بدايته ونهايته. مثال: (بين 5 و 30: 30 غير محتسبة من ضمن المجال)	
<pre>import random x= random.randrange(5, 30, 2) print(x) Result may be: 5 or 7 or 9 or 11 or etc</pre>	5- لاختيار عدد عشوائي صحيح ضمن مجال تحدّد بدايته ونهايته وخطوته.	

	مثال: (بين 5 و30 خطوة 2: 30 غير محتسبة من ضمن المجال)	
txt="Welcome to Bahrain" words=txt.split() print(words) Result: words=['Welcome','to','Bahrain']	لتقسيم نص إلى كلمات	split()
cars=[Math, 'Science', 'English'] cars.pop(0) print(cars) Result is: ['Science', 'English']	لحذف عنصر من سلسلة بناء على موقعه .index	pop()
	لحذف كائن بأكمله (سلسلة أو متغير أو جزء من سلسلة)	del()

نشاط 1

inputs	instructions	results
x=-2	y=abs(x)	y=
a=[2,5,9,10,5,6,7,8,9]	b = len(a)	b=
z="123"	n = int(z)	n=
z=123	s=str(z)	s=
r=range(6)		r contains ()
r=range(0,6,2)		r contains ()
n=4	f = float(n)	f=

name="othman"	l=len(name)	l=
L=[5,6,2,0,7]	M=min(L)	M=
L=[5,6,2,0,7]	M=max(L)	M=
L=[5,6,2,0,7]	M=sum(L)	M=
L=[5,6,2,0,7]	M=sorted(L)	M=[]
L=[5,6,2,0,7]	M=sorted(L, reverse=true)	M=[]
L=[5,6,2,0,7]	i=L.index(0)	i=
L=[5,6,2,0,7]	L.remove(5)	L=[]
L=[5,6,2,0,7]	L.pops(3)	L=[]
r is a variable	r=random.random()	r is between ____ and ____
r is a variable	r=random.randint(15,20)	r is between ____ and ____
r is a variable	r=random.randrange(15,20)	r is between ____ and ____
r is a variable	r=random.randrange(15,25,5)	r is between ____ and ____

How many shortest-length paths are there to get from your house to the doughnut shop?

↑ up
→ right

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$

$\binom{11}{7} = \binom{11}{4} = 330$ paths

There are six dogs to give use a "stars and bars" diagram to illustrate and sixth dog gets 3 tacos, the second dog the third dog gets 5 and the fourth dog 0

☆☆☆||☆☆☆☆☆|☆||☆☆

$A = \{2, 4, 10, 14\}$

$(n) = \frac{n!}{(n-k)!k!}$

P	Q	R	P ∨ Q	P ∨ R	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	T	F	F	F	F
F	F	T	F	T	F
F	F	F	F	F	F

$7, 11, 15, 19, 23, \dots$

$A_1 - A_2 = 4$
 $A_2 - A_3 = 4$
 $A_3 - A_4 = 4$
 $\vdots$
 $A_{n-1} - A_n = 4$
 $A_n - A_{n+1} = 4$

$A = \{2, 4, 10, 14\}$

الدرس العاشر

الجملة الشرطية

CONDITIONAL STATEMENT

أهداف الدرس



يكون الطالب في نهاية هذا الدرس قادراً على:

- 1- التعرف على بنى الجمل الشرطية (بسيطة - كاملة - متداخلة)
- 2- توظيف الجملة الشرطية المناسبة لحل مشكلة بومجية ما.
- 3- يبني اتجاهات ايجابية نحو الجمل الشرطية.

■ الجمل الشرطية Conditional Statements

○ الجملة الشرطية البسيطة:

PSEUDOCODE

IF الشرط THEN الاجراءات ENDIF

تمكّني هذه الجملة البرمجية من تنفيذ إجراءات معيّنة بناء على تحقق شرط محدّد من عدمه.

مثال 1

أريد أن يعرض الحاسوب الرقم الذي أقوم بإدخاله بشرط أن يكون رقما فرديا (مثل 1 - 3 - 5).

الحل

INPUT number

IF number % 2 == 1

THEN

OUTPUT (number)

ENDIF

باقي القسمة على 2 عندما = 1
يعني أنّ القاسم هو عدد فردي.

PYTHON

بنية الجملة الشرطية البسيطة:

if condition :

↔ statement1

↔ statement2

↔

يمثّل الرمز (:) في
الجملة الشرطية
المصطلح THEN

جميع الجمل البرمجية التي تقع أسفل الجملة الشرطية وعلى
مسافة من السطر تكون تابعة لها.
الرجوع إلى بداية السطر يعني نهاية الجملة الشرطية

يكون حلّ المثال أعلاه بلغة Python كالآتي:

```
File Edit Format Run Options Window Help
1 number=input("Plz enter an integer:")
2 if (int(number)%2==1):
3     print(number)
```



1. أكتب بلغة **Pseudocode**، برنامجاً يمكّنك من إدخال المعدّل الفصلي للطالب وإظهار الرسالة "ناجح" إذا كان المعدّل يساوي 60 فما فوق.
2. حوّل الكود في السؤال الأول إلى كود بلغة Python.

○ الجملة الشرطية الكاملة:

PSEUDOCODE

IF الشرط **THEN**

الإجراءات إذا تحقّق الشرط

ELSE

الإجراءات إذا لم يتحقّق الشرط

ENDIF

تمكّني هذه الجملة الشرطية من تنفيذ إجراءات معيّنة إذا تحقّق الشرط وإجراءات أخرى إذا لم يتحقّق.

أريد أن يستقبل الحاسوب رقما صحيحا ثم يظهر الرسالة "هذا الرقم زوجيًا" إذا كان زوجيًا أو الرسالة "هذا الرقم فرديًا" إذا كان فرديًا.

```
INPUT ("Plz Enter an integer:", number)
IF number % 2 == 0
    THEN
        OUTPUT ("هذا الرقم زوجي")
    ELSE
        OUTPUT ("هذا الرقم فردي")
ENDIF
```

PYTHON

بنية الجملة الشرطية الكاملة في لغة Python:

if *condition* :

↔ statement1

↔ statement2

↔

else:

↔ statement1

↔ statement2

↔

ألاحظ أن المصطلحات
المفتاحية **if** و **else**
موجودان على نفس
المستوى العمودي.

يكون حلّ المثال أعلاه بلغة Python كالآتي:

```
File Edit Format Run Options Window Help
1 number=input("Plz enter an integer:")
2 if (int(number)%2==0):
3     print("هذا الرقم زوجي")
4 else:
5     print("هذا الرقم فردي")
6
```

نشاط 2

1. أكتب بلغة **Pseudocode** برنامجاً يمكنك من عرض نتيجة عملية قسمة لعددتين صحيحين ويتأكد قبل إجراء العملية من:
 - أ. إذا كان العدد المقسوم عليه يساوي صفراً، يظهر الرسالة ("لا يمكن القسمة على صفر").
 - ب. إذا كان العدد المقسوم عليه لا يساوي صفراً، يعرض نتيجة القسمة.

2. حوّل الحلّ من pseudocode على لغة Python.
3. غير الشرط بحيث يكون المقسوم عليه لا يساوي صفراً

نشاط 3

1. أكتب بلغة **Pseudocode** برنامجاً يمكنك من حساب سعر تذكرة الدخول إلى حلبة سباق البحرين للفورمولا 1 بناء على المنطقة وعمر الشخص بحيث يتمتع الشخص الذي يبلغ من العمر 18 سنة أو أقلّ بتخفيض 25%.

المعطيات:

- المنطقة A: سعر التذكرة 35 دب
 - المنطقة B: سعر التذكرة 50 دب
 - المنطقة VIP: سعر التذكرة 100 دب
2. حوّل الحلّ من pseudocode على لغة Python.

○ الجملة الشرطية المتداخلة:

PSEUDOCODE

IF الشرط THEN

الإجراءات إذا تحقق الشرط الأول

ELSEIF

الإجراءات إذا لم يتحقق الشرط الثاني

ELSE

الإجراءات إذا لم يتحقق أي من الشروط أعلاه

ENDIF

مثال 3

في قانون الطرقات هناك أربع حالات للإشارة الضوئية:

- 1- **الأحمر**: يجب عليك التوقف
- 2- **الأخضر**: السماح لعبور التقاطع.
- 3- **الأخضر** المتقطع: الانتباه والمرور إذا كانت الإشارة لا تزال في حالة وميض.
- 4- **الأصفر**: الاستعداد للتوقف ويفضل عدم المجازفة وقطع الطريق في هذا اللون.

سأكتب كوداً مزيّفاً لإظهار رسالة لكل حالة من هذه الحالات.

OUTPUT ("اختر الرقم حسب حالة الإشارة الضوئية")

OUTPUT ("1-red 2-green 3-blinking green 4-Yellow")

INPUT ("أدخل رقما من 1 إلى 4", sign)

IF sign = 1 THEN // الشرط الأول

OUTPUT("لا يسمح لك بالعبور")

ELSEIF sign = 2 THEN // الشرط الثاني

OUTPUT ("يسمح لك بالعبور")

ELSEIF sign=3 THEN // الشرط الثالث

OUTPUT ("انتبه واعبر")

ELSEIF sign = 4 THEN // الشرط الرابع

OUTPUT ("خفف السرعة واستعد للوقوف")

ELSE // كل الشروط السابقة لم تتحقق

OUTPUT ("الرقم المدخل خطأ")

ENDIF

PYTHON

بنية الجملة الشرطية المتداخلة في لغة Python

if *condition1*:

↔ *statement1*

↔ *statement2*

↔

elif *condition2*:

↔ *statement1*

↔ *statement2*

↔

else:

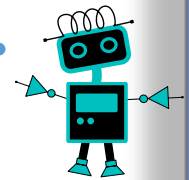
↔ *statement1*

↔ *statement2*

↔

ألاحظ أنَّ المصطلحات
المفتاحية **if** و **elif** و
else موجودان على نفس
المستوى العمودي.

```
File Edit Format Run Options Window Help
1 print("اختر الرقم حسب حالة الإشارة الضوئية")
2 print("1-red 2-green 3-blinking green 4-Yellow")
3 sign=input("أدخل رقماً من 1 إلى 4:")
4
5 if int(sign)==1:
6     print("لا يسمح لك بالعبور")
7 elif int(sign)==2:
8     print("يسمح لك بالعبور")
9 elif int(sign) == 3:
10    print("التب واعد")
11 elif int(sign)==4:
12    print("خفف السرعة واستعد للوقوف")
13 else:
14    print("الرقم الذي أدخلته خاطئ")
15
```



يكون حلّ المثال
أعلاه بلغة
Python كالآتي:

أرغب في كتابة برنامج بلغة Python لعرض الوصف المناسب لشخص ما حسب عمره بناء على الشروط الآتية:

الوصف	الشرط
رضيع	أصغر من 2 سنة
طفل	أصغر من 12 سنة
مراهق	أصغر من 18 سنة
شاب	أصغر من 30 سنة
كهل	أصغر من 60 سنة
شيخ	أخرى



1- هل الخريطة التدفقية أمامك تمثل الحل المناسب؟

2- أجب عن الآتي:

a. ما هي نتيجة إسناد القيمة 17 إلى المتغير age؟

b. في حال عكسنا علامة المقارنة إلى أكبر من، ماذا ستكون النتيجة؟

c. برّر سبب الخطأ.

3- ما هو الحل الذي تراه مناسباً؟

4- قم بتصميم الخريطة التدفقية الصحيحة.

LOOP STATEMENT

3- يبنى اتجاهات ايجابية نحو الجمل التكرارية.

■ الجملة التكرارية loop Statements

○ الجملة التكرارية for loop:

PSEUDOCODE

```
FOR counter = start TO end
    instructions...
ENDFOR
```

تمكّني هذه الجملة البرمجية من تكرار تنفيذ إجراءات معينة عدد محدّد من المرات.

مثال 1

أريد أن يعرض الحاسوب الأرقام من 5 إلى 10.

الحل

```
FOR c = 5 to 10
    OUTPUT c
ENDFOR
```

PYTHON

```
for counter in range() || string || list:
    instructions...
```

بنية الجملة التكرارية for loop:

جميع الأوامر البرمجية التي تقع أسفل الجملة التكرارية وعلى مسافة من السطر تكون تابعة لها.
الرجوع إلى بداية السطر يعني نهاية الجملة التكرارية

يكون حلّ المثال أعلاه بلغة Python كالآتي:

```
for c in range(5,11):
    print(c)
```

OUTPUT:

5
6
7
8
9
10

ألاحظ في الدالة range() أن نهاية المجال تكون دائما 1+ عن الحد الأقصى المطلوب

تعبئة سلسلة list:

أعلمُ أنَّ السلسلة list مكوّنة من مجموعة عناصر يتم إدخالها بالتعيين المسبق. ماذا لو أردت إدخال عناصر السلسلة list أثناء تنفيذ البرنامج؟ لإنجاز هذا أحتاج إلى:

- عدد عناصر السلسلة التي أريد إدخالها.
- الجملة التكرارية for ليكون إدخال العناصر متتاليا.
- الدالة append التي ستمكنني من إضافة العنصر إلى السلسلة.

المطلوب:

1. تعبئة سلسلة list باسم ls بـ 3 أعداد صحيحة.
2. فيكون كود البرنامج كالآتي:

```
File Edit Format Run Options Window Help
ls=[] # يجب تعريف المتغير قائمة |
for i in range(3): # تحديد المجال
    item=input("enter an integer:") # إدخال عدد صحيح
    ls.append(int(item)) # إضافة العدد الصحيح إلى القائمة
print("The list entered is:",ls) # عرض القائمة بعد إدخالها
```

```
enter an integer:1
enter an integer:2
enter an integer:3
The list entered is: [1, 2, 3]
```

3- النتيجة بعد تنفيذ البرنامج:

1. صمّم الخريطة التدفقية **Flowchart**، تمكّنك من عرض مضاعفات الرقم 3 الآتية:

OUTPUT:

3
6
9
12

2. حوّل الحلّ إلى لغة Python.

○ الأمر **break**

يمكنني الأمر **break** من كسر حلقة التكرار بناء على شرط محدد.

مثال 2

ضمن السلسلة **nbr** الآتية:

`nbr = [1,2,3,4,5]`

أرغب في عرض محتوى المتسلسلة **nbr** شرط التوقف عند الرقم 4 (دون عرض الرقم 4).

الأوامر البرمجية بلغة Python:

```
nbr = [1,2,3,4,5]
for x in nbr:
    if x == 4:
        break
    print(x)
```

النتيجة:

1
2
3

○ الأمر **continue**

يمكّنني هذا الأمر من تجاوز عنصر محدّد ضمن جملة تكرارية.

النتيجة:

```
nbr=[1,2,3,4,5]
for x in nbr:
    if x==4:
        continue
    print(x)
```

1
2
3
5

نشاط 2

لنعتبر السلسلة s3 الآتية:

`s3 = [6, 5, 12, 7, 9]`

أكتب برنامجاً بلغة **Python** يمكنك من حساب مجموع الأعداد ضمن السلسلة s3 التي تقبل القسمة على **3** فقط.
في مثالنا هذا نتوقع أن تكون النتيجة:

$6 + 12 + 9 = 27$

ملاحظة: استعن بالأمر `continue` لإيجاد الحل.

نشاط 3

لنعتبر السلسلة الآتية:

`names = ['ahmed', 'salah', 'Ali', 'ridha', 'amal']`

أكتب برنامجاً بلغة **Python** يعرض الأسماء التي تبدأ بحرف "a" أو "A" فقط.

نشاط 4

لنعتبر السلسلتان الآتيتان:

`Races = ['BahrainGP', 'SaudiGP', 'AustralianGP', 'ChineseGP', 'AzerbaijanGP']`

`Dates = ['2023-03-05', '2023-03-19', '2023-04-02', '2023-04-16', '2023-04-30']`

تمّ إلغاء سباق "**ChineseGP**" الذي كان سيقام في تاريخ "**2023-04-16**"، أكتب برنامجاً بلغة **Python** يمكنك من:

1- حذف السباق الملغى من السلسلتين "**Races**" و "**Dates**".

2- عرض كل سباق متبوعاً بتاريخه.

○ الجملة التكرارية while loop:

PSEUDOCODE

```
WHILE condition DO
    instructions...
ENDWHILE
```

تستمرّ جملة التكرار while في تنفيذ الأوامر الموجودة ضمنها طيلة تحقّق الشرط *condition*.

مثال 3

أريد أن أستمّر في إدخال بيانات عدديّة حتى إدخال الرقم 0.

الحل

```
WHILE n<>0
    INPUT ("Enter a number",n)
ENDWHILE
```

PYTHON

```
while condition :
    instructions...
```



جميع الأوامر البرمجية التي تقع أسفل الجملة التكرارية وعلى مسافة من السطر تكون تابعة لها.
الرجوع إلى بداية السطر يعني نهاية الجملة التكرارية

يكون حلّ المثال أعلاه بلغة **Python** كالآتي:

```
File Edit Format Run Options Window Help
n=1
while int(n) != 0:
    n=input("Enter a number:")
```

أجب:

- (1) $n=1$: لماذا أسندت قيمة إلى المتغير n قبل بدء الجملة التكرارية؟
- (2) لماذا استعنت بالدالة **int()** ما دمت قد أسندت قيمة رقمية إلى المتغير n في بداية البرنامج؟

نشاط 5

أريد أن يستمرّ الحاسوب في حساب مجموع أعداد مختلفة مدخلة حتى يفوق مجموعها الـ 100 ثمّ يعرض المجموع.

1- أكتب الحلّ بلغة Pseudocode.

2- ارسم الخريطة التدفقية للحلّ.

3- حوّل الحلّ إلى لغة Python.

هام جداً: على خلاف بقية لغات البرمجة تمكّني لغة Python من إرفاق الجملة التكرارية **while** بالخاصية **else** التي نرفقها عادة بالجملة الشرطية.

يتبع المثال السابق:

أريد أن أستمّر في إدخال بيانات عددية حتى إدخال الرقم 0 وحينها تظهر الرسالة: "شكرا لقد أتممت إدخال الأعداد المطلوبة".

```
File Edit Format Run Options Window Help
n=1
while int(n) != 0:
    n=input("Enter a number:")
else:
    print("شكرا لقد أتممت إدخال الأعداد المطلوبة")
```

Result is:

```
Enter a number:20
Enter a number:12
Enter a number:0
شكرا لقد أتممت إدخال الأعداد المطلوبة
```

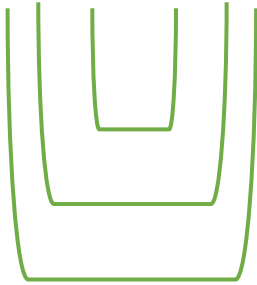
أرغب في برمجة لعبة بسيطة تتمثل في:

- 1- يختار الحاسوب عددا عشوائيا صحيحا ضمن مجال تحدده. (راجع الدالة `random.randint(start, end)`)
- 2- أخمن العدد الذي اختاره الحاسوب من خلال إدخال قيمة في متغير أعينه.
- 3- إذا كانت القيمة التي أدخلتها أصغر من القيمة العشوائية تظهر الرسالة ("هذه القيمة أصغر من العدد المجهول")
- 4- إذا كانت القيمة التي أدخلتها أكبر من القيمة العشوائية تظهر الرسالة ("هذه القيمة أكبر من العدد المجهول")
- 5- إذا أدخلت القيمة الصحيحة تظهر الرسالة ("ممتاز تخمينك صحيح.") ويظهر العدد المجهول.
- 6- أكتب الحل بلغة Python موظفا الجملة الشرطية والجملة التكرارية المرفقة بخاصية `else`.

أكتب برنامجاً بلغة Python يمكنك من:

- 1- إدخال كلمة ضمن متغير باسم "word".
- 2- التأكد من أن هذه الكلمة متناظرة أو لا.
- 3- الكلمة المتناظرة هي التي تقرأ من الاتجاهين. (racecar – madam – civic - ...etc)

racecar



أكتب حلاً برمجياً لحذف القيم المكررة من سلسلة list. متبعا التوجيهات الآتية:

- 1- يتم إدخال 20 عدد صحيح عشوائيا ضمن السلسلة list.
- 2- أكتب الحل عن طريق الكود المزيّف أولا.
- 3- حوّل الكود إلى لغة Python.

أكتب حلاً برمجياً لتقسيم نص إلى كلمات يتم حفظها في سلسلة list:

- 1- يتم إدخال النص عن طريق لوحة المفاتيح وحفظه في متغير.
- 2- النص لا يحتوي على علامات ترقيم.

CREATE FUNCTIONS

3- توظيف الدوال ضمن الحلول البرمجية.

■ ما هي الدوال؟

الدالة هي كتلة من الجمل البرمجية الجاهزة مسبقاً، يتم استدعاؤها عند اللزوم، عادة ما ترافقها عوامل خاصّة بها تسمى "parameters" لاستخدامها للوصول إلى النتيجة المطلوبة. عادة ما تقوم الدالة بإرجاع النتيجة إلى البرنامج الرئيسي الذي تم استدعاؤها منه.

كما أنّها تساهم في حلّ المسائل المعقّدة بحيث يتمّ تقسيم المشكل الرئيسي إلى مشكلات بسيطة يتمّ حلّها بطريقة منفصلة ثمّ تجميعها للحصول على الحلّ النهائي.

PSEUDOCODE

1- قالب العام للدالة ضمن الكود المزيّف PSEUDOCODE:

```
FUNCTION function_name (par1, par2, ...)
```

```
    Instructions
```

```
    .....
```

```
END FUNCTION
```

PYTHON

2- قالب العام للدالة ضمن لغة البرمجة PYTHON:

```
def function_name(parameters):
```

```
    instructions
```

```
    .....
```

أ. دالة بدون إرجاع نتيجة:

أ. يبدأ بناء الدالة بـ **def**

أ. اسم الدالة

أ. متغيرات الدالة إن وجدت

أ. ترك مسافة بادئة وكتابة الأوامر *instructions*.

سأنشئ دالة تقوم بإظهار رسالة فقط ولا أحتاج منها إرجاع أيّة نتيجة.

```
File Edit Format Run Options Window Help
def msg():
    print("This is a function without parameters.")
```

- لكي تنفّذ هذه الدالة سأقوم باستدعائها كالآتي:

```
File Edit Format Run Options Window Help
def msg():
    print("This is a function without parameters.")

#call of the function msg()

msg()
```

استدعاء الدالة يأتي
بعد تعيينها.

- أنفّذ البرنامج فأتحصّل على النتيجة الآتية:

```
This is a function without parameters.
```

ب. دالة مع إرجاع نتيجة:

```
File Edit Format Run Options Window Help
def sumTwo(a, b):
    s=a+b
    return s
#### نهاية الدالة ####
```

1- سأنشئ دالة لحساب مجموع عددين.

2- عدد عوامل الدالة parameters هو 2 (a و b) وهو عدد الأعداد التي سأجمعها.

3- return s: تمثّل نتيجة الدالة التي

ستحوّل إلى الجزء الرئيسي من البرنامج.

```
##### الجزء الرئيسي من البرنامج #####
x=5
y=4
z=sumTwo(x, y) # استدعاء الدالة
print(z)
```

4- استدعاء الدالة
من الجزء الرئيسي
من البرنامج.

5- يجب استدعاء
الدالة بنفس عدد

العوامل parameters التي أنشأت بها وبنفس الترتيب.

6- يتم تعويض عوامل الدالة الرئيسية بالعوامل التي استدعيت بها.

7- أنفذ البرنامج وألاحظ النتيجة ← ستكون النتيجة = 9

مثال 3

1- سأنشئ دالة لحساب كتلة الجسم:

أ. سأحتاج إلى متغير الطول height بالمتر.

ب. سأحتاج إلى الوزن weight بالكيلو.

ج. معادلة حساب كتلة الجسم BodyMass =

$$\text{BodyMass} = \text{weight} / (\text{height} * \text{height})$$

2- إنشاء الدالة:

```
File Edit Format Run Options Window Help
def bodym(h, w):
    b = w / (h*h)
    return b
```

3- الجزء الرئيسي من البرنامج:

```
### الجزء الرئيسي من البرنامج ###
weight = input("Enter your weight in kg:")
height = input("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)
```

4- لا أنسى أن القيم المدخلة عن طريق `input` تكون من نوع نصي، لذلك قمت بتحويلها عند استدعاء الدالة (`height` من نوع عشري لذلك حولته إلى `float` و `weight` من نوع عدد صحيح فحوّلته إلى `int`)

5- إذا علمت أنّ:

أ. الوزن المثالي يكون بين 18.5 و 24.9

ب. مرحلة السمنة من 25.0 وما فوق

ج. مرحلة سوء التغذية من 18.4 وأقلّ

6- أنشئ، أسفل الدالة الأولى، دالة جديدة تسمى `result` تظهر وصف نتيجة حساب كتلة الجسم وفق المعايير أعلاه.

```
def result(bm):
    if bm>=18.5 and bm < 25.0:
        print("excelent")
    elif bm>=25.0:
        print("Overweight")
    else:
        print("Underweight")
```

7- استدعاء دالة `result` يكون بعد استدعاء دالة `bodym`.

```
### الجزء الرئيسي من البرنامج ###
weight = input("Entyer your weight in kg:")
height = input ("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)
```

8- البرنامج كاملاً:

```

File Edit Format Run Options Window Help
def bodym(h,w):
    b = w/(h*h)
    return b

def result(bm):
    if bm>=18.5 and bm < 25.0:
        print("excellent")
    elif bm>=25.0:
        print("Overweight")
    else:
        print("Underweight")

### الجزء الرئيسي من البرنامج ###
weight = input("Entyer your weight in kg:")
height = input ("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)

```

مثال 4

- 1- سأنشأ دالة باسم **CreateList** لإدخال عناصر سلسلة **list** مكونة من أعداد صحيحة تُحدّد عددها.
- 2- سأنشأ دالة باسم **SumList** لحساب مجموع عناصر السلسلة.
- 3- إظهار النتيجة.

```

File Edit Format Run Options Window Help
1 ##### Create the list #####
2 def CreateList():
3     ls=[]
4     x=input("Length of the list:")
5
6     for i in range(int(x)):
7         item=input("Enter an item of the list:")
8         ls.append(int(item))
9
10    return ls
11
12 ##### Sum of the list #####
13 def SumList(ls):
14     sum=0
15     for i in range(len(ls)):
16         sum=sum+ls[i]
17     print("The sum is:",sum)
18
19 ##### Main #####
20 ls=CreateList()
21 print(ls)
22 SumList(ls)

```

نشاط 1

أكتب برنامجاً بلغة Python يمكّنك من برمجة نموذج لآلة حاسبة متّبعاً الخطوات الآتية:

- 1- يتمّ إدخال العددين اللذين سيتمّ إجراء العملية الحسابية عليهما من خلال الجزء الرئيسي من البرنامج.
- 2- يتمّ إدخال المعامل الحسابي المطلوب: (+ أو - أو / أو *)
- 3- في حال إدخال علامة الجمع: يتمّ استدعاء دالة الجمع وتنفيذ العملية وإرجاع النتيجة إلى البرنامج الرئيسي.
- 4- في حال إدخال علامة الطرح: يتمّ استدعاء دالة الطرح وتنفيذ العملية وإرجاع النتيجة إلى البرنامج الرئيسي.
- 5- وكذلك الحال لباقي العوامل الحسابية.
- 6- يتمّ عرض نتيجة العملية الحسابية من البرنامج الرئيسي من خلال القيمة المُرجّعة من الدالة المعنية.

نشاط 2

- 1- عيّّن ضمن سلسلة list مجموعة هامة من الكلمات.
- 2- اختر كلمة عشوائية من القائمة.
- 3- ادخل حرفاً من الحروف.
- 4- أظهر عدد مرات تواجد هذا الحرف في الكلمة ومواقعها.
- 5- حلّ المسألة مستعيناً بالدوال والخوارزميات اللازمة.

مثال: الكلمة: manama

الحرف: a

النتيجة: عدد مرات تواجد الحرف a ضمن الكلمة manama = 3

المواقع index: (2، 4، 6)

الدرس الثالث عشر

بايثون والذكاء الاصطناعي

Python & AI

أهداف الدرس

يكون الطالب في نهاية هذا الدرس قادراً على:

1- تعريف الذكاء الاصطناعي.

2- يبي أهمية الذكاء الاصطناعي.

3- يتعرف أهمية التعلم الآلي وعلاقته بالذكاء الاصطناعي.

4- القدرة على تحديد المخرجات المطلوبة في بعض المجالات الاقتصادية من خلال

قاعدة بيانات ينشئها ويستدل بها بواسطة الذكاء الاصطناعي إلى أفضل الخيارات المناسبة

لهذا المجال الاقتصادي.

مفهوم الذكاء الاصطناعي

يعرف الذكاء الاصطناعي بأنه الذكاء الذي تبديه الآلات والبرامج بما يحاكي القدرات الذهنية البشرية وأنماط عملها، مثل القدرة على التعلم والاستنتاج ورد الفعل على أوضاع لم تبرمج في الآلة.

الهدف من الذكاء الاصطناعي

إنّ الهدف الرئيسي من الذكاء الاصطناعي يكمن في اتخاذ القرار الصحيح في أقل وقت وبأقل التكاليف ممّا سيساعد المؤسسات على:

- الزيادة في الإنتاج
- التنبؤ للتهديدات الأمنية المستقبلية
- تقليل التكلفة
- ربح الوقت
- سرعة اتخاذ القرار

كما يمكن أيضا الاستفادة من الذكاء الاصطناعي في مجال التعليم من خلال تحليل البيانات وتحديد الموضوعات والدروس التي يجب إعادة تقييمها ووضع أفضل برنامج تعليمي للطلاب.

ما لمقصود بالتعلم الآلي Machine Learning



على سبيل المثال: يمكن لعلماء البيانات تدريب تطبيق طبي على تشخيص مرض السرطان من صور الأشعة السينية عن طريق تخزين ملايين الصور المفحوصة والتشخيصات المقابلة.

ما أهمية التعلم الآلي؟

يساعد التعلم الآلي الأعمال عن طريق دفع عجلة النمو وفتح سبل إيرادات جديدة وحل مشكلات صعبة. فالبيانات هي القوة المحركة الحاسمة وراء صنع القرار التجاري، ولكن في الأحوال التقليدية، كانت تستخدم الشركات البيانات من مصادر مختلفة، مثل ملاحظات العملاء والموظفين والوضع المالي. وتعمل أبحاث التعلم الآلي على أتمتة هذه العملية وتحسينها. عن طريق استخدام البرمجيات التي تحلل كميات هائلة من البيانات بسرعات عالية، تستطيع الأعمال تحقيق النتائج بصورة أسرع.

أين يُستخدم التعلم الآلي؟

تطبيقات التعلم الآلي في بعض المجالات الأساسية:



<https://aws.amazon.com/ar/what-is/machine-learning/>

يخمن، من خلال تحليل كمّ كافٍ من البيانات، علاقة رياضية بين المدخلات والمخرجات

1

كلّ لوغاريتم للتعلم الآلي يبني على أساس دالة رياضية قابلة للتعديل.

2

ويمكن فهم المبدأ الأساسي هكذا:

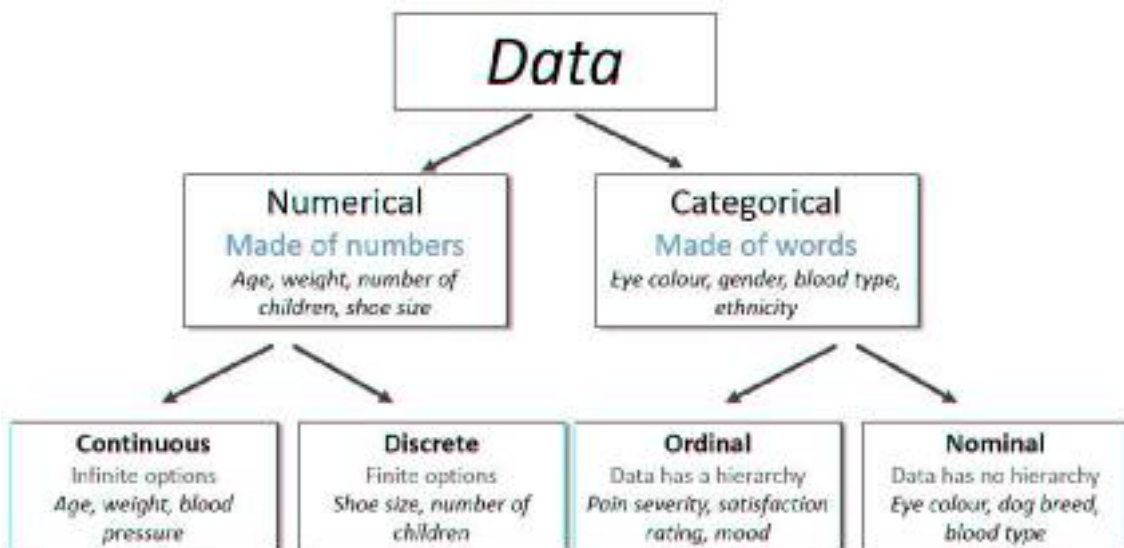
نحن "ندرب" اللوغاريتم عن طريق تزويده بمجموعات المدخلات/المخرجات (input,output) التالية – (2,10)، و(5,19)، و(9,31)

يحسب اللوغاريتم العلاقة بين المدخلات والمخرجات لتكون $output = 3 * input + 4$:

ثم نعطيه المدخل 7، ونطلب منه أن يتوقع المخرج. وهو يستطيع تحديد أن المخرج هو 25.

من خلال هذا الفهم الأساسي، يركز التعلم الآلي في المبدأ الذي ينص على أن كل نقاط البيانات المعقدة يمكن ربطها رياضياً عن طريق أنظمة الحاسوب ما دامت مزودة بالقدر الكافي من البيانات وقدرة الحوسبة على معالجة هذه البيانات. لذلك، دقة المخرج تربطه علاقة طردية بحجم المدخل المُعطى.

أنواع البيانات:





التعلم الآلي الخاضع للإشراف (Supervised Learning)

في هذا النوع من التعلم الآلي، يقوم المبرمجون بتزويد الخوارزميات ببيانات محددة ومعرفة، ويتم تحديد المتغيرات التي يرغبون من الخوارزمية تقييمها. يعني هذا الأمر أن المدخلات والمخرجات المطلوبة محددة سلفاً.

التعلم غير الخاضع للإشراف (Unsupervised Learning)

يتضمن هذا النوع من التعلم الآلي على خوارزميات يتم تزويدها ببيانات غير معرفة أو مسماة. وتقوم الخوارزمية بمسح البيانات بحثاً عن أي اتصال أو ترابط فيما بينها. لكن مثل النوع السابق، تكون البيانات والتنبؤات محددة سلفاً.

التعلم شبه الخاضع للإشراف (Semi-Supervised Learning)

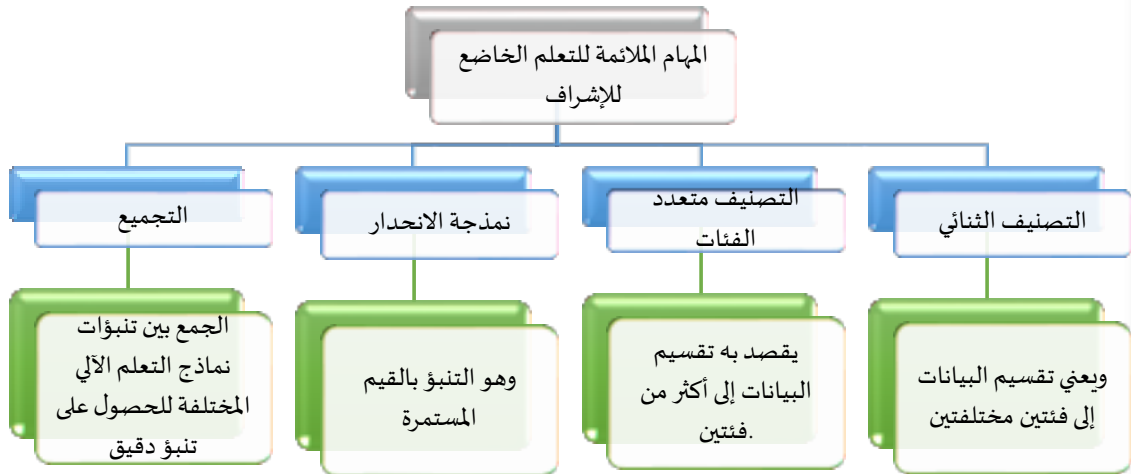
يتضمن هذا النوع مزيجاً من النوعين السابقين، حيث يقوم المبرمجون بتغذية الخوارزميات بالبيانات، لكن تترك الحرية للخوارزميات لاستكشاف البيانات بنفسها وتطوير فهمها الخاص بهذه البيانات.

التعلم المعزز (Reinforcement Learning)

عادة ما يتم استخدام التعلم المعزز لتعليم الآلية إكمال عملية متعددة الخطوات ضمن قواعد واضحة ومحددة. حيث تقرر الخوارزميات الخطوات التي يجب اتخاذها لتحليل البيانات من تلقاء نفسها.

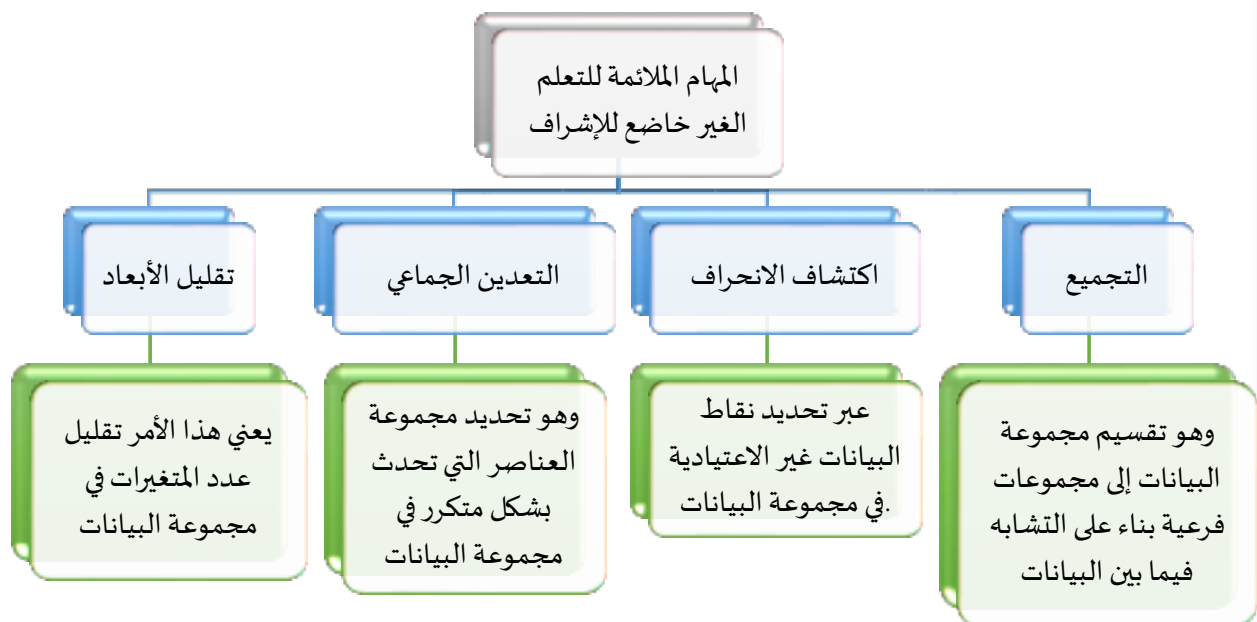
كيف يعمل التعلم الخاضع للإشراف؟

يتطلب تعلم الآلة الخاضع للإشراف من المبرمج تدريب الخوارزمية على كيفية التعامل مع المدخلات الموجودة والمخرجات المرغوب بها.



كيف يعمل التعلم غير الخاضع للإشراف؟

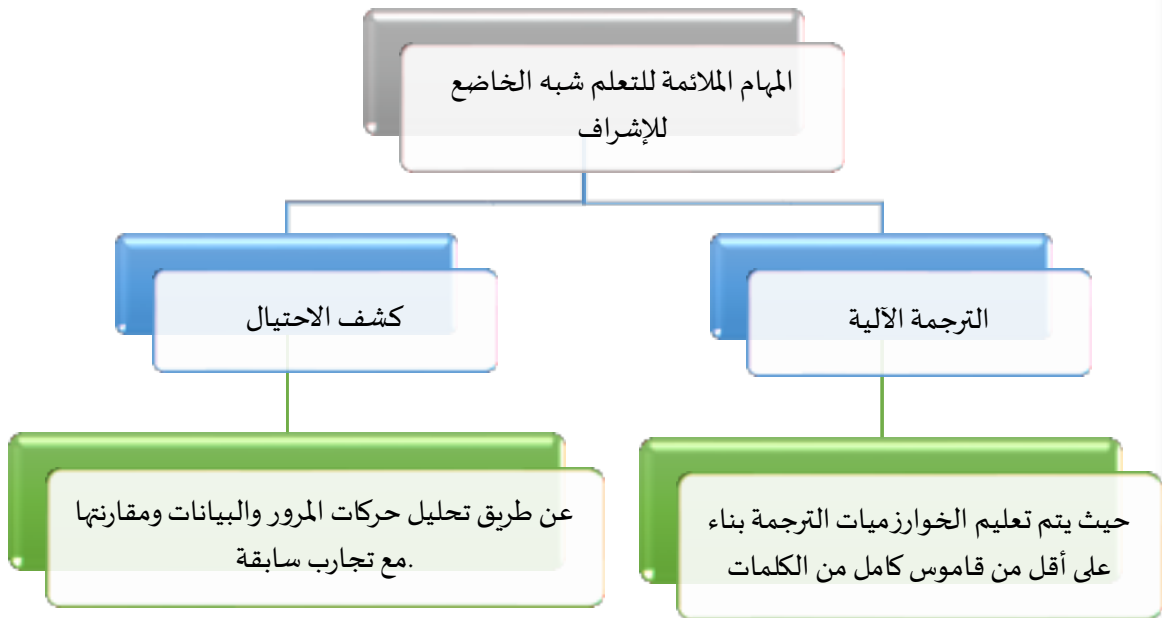
في النظام غير الخاضع للإشراف، لا تحتاج الخوارزميات إلى تسمية أو تعريف البيانات، حيث أنها تقوم بفحص البيانات غير المعرفة للبحث عن الأنماط التي يمكن استخدامها لتجميع البيانات في مجموعات. معظم أدوات التعلم العميق، بما في ذلك الشبكات العصبية، هي خوارزميات غير خاضعة للإشراف.



كيف يعمل التعلم شبه الخاضع للإشراف؟

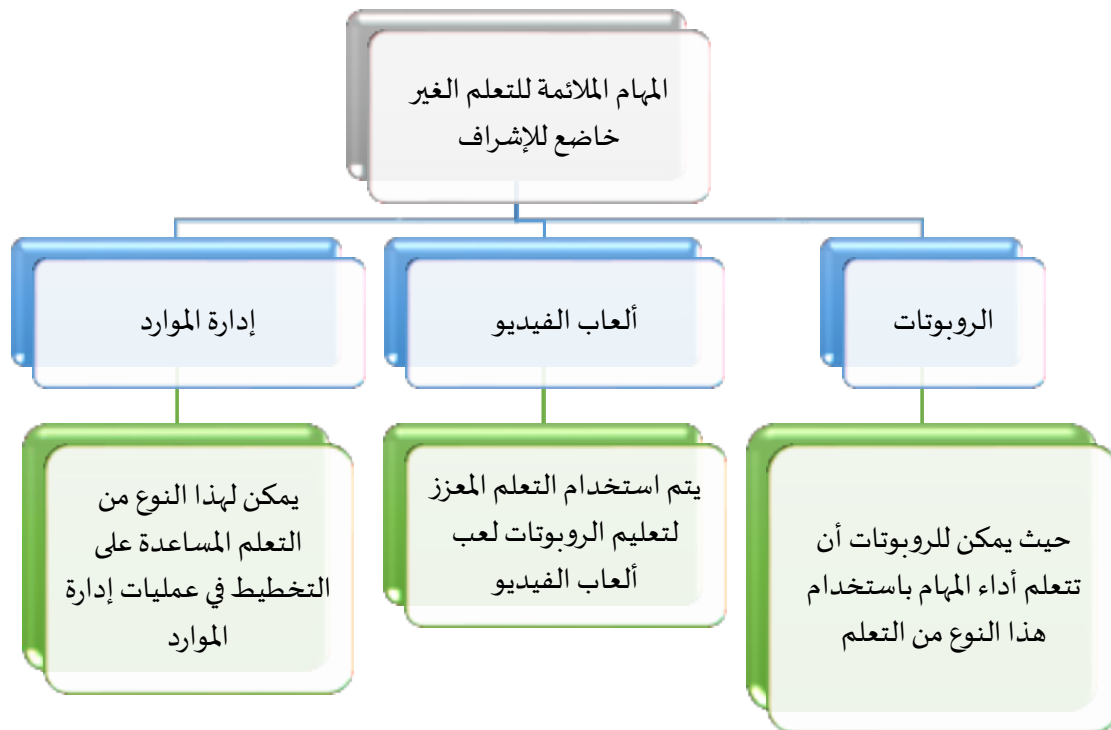
يعمل هذا النوع من التعلم عن طريق تزويد الخوارزميات بمجموعة صغيرة من بيانات التدريب، حيث أن هذه البيانات تكون معرفة. عن طريق هذا الأمر، تتعلم الخوارزمية الأبعاد المتعلقة بهذه البيانات، وتتمكن بعد ذلك من تطبيقها على البيانات الجديدة غير المعرفة التي تحصل عليها.

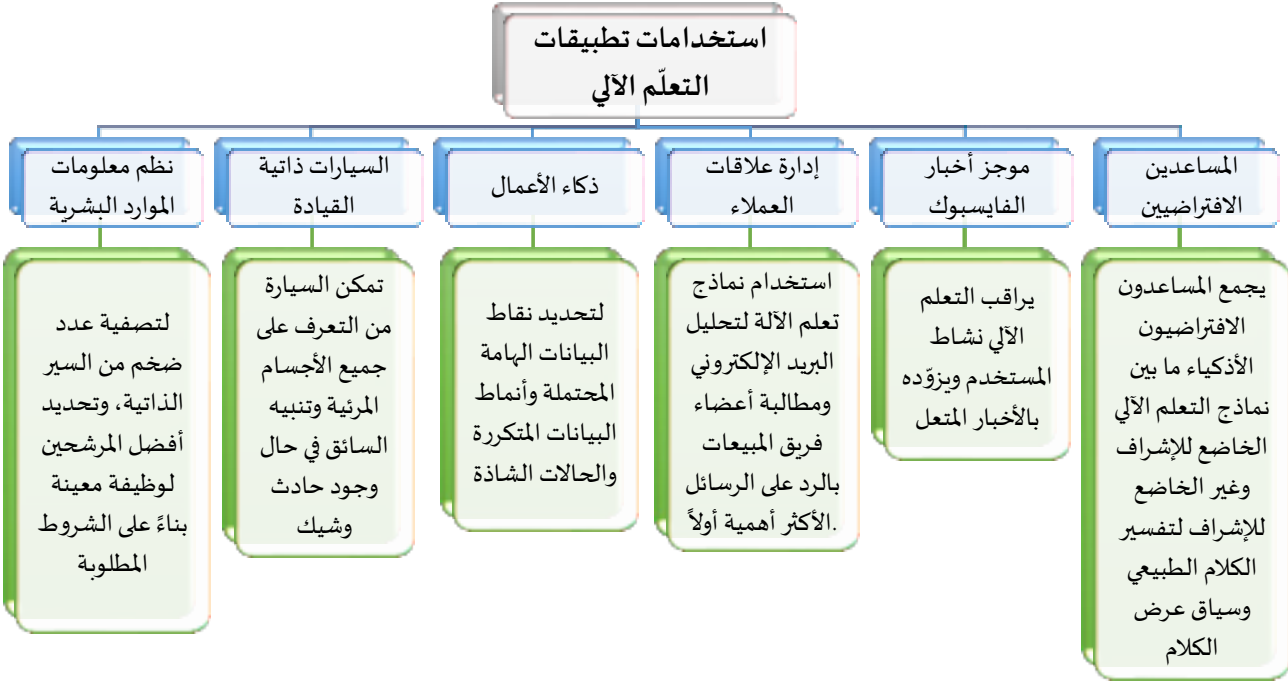
وعادة ما يتحسن أداء الخوارزميات عندما يتم تدريبها على مجموعات متعددة من البيانات المعرفة والمصنفة. لكن قد يكون تصنيف البيانات مكلفاً من ناحية مادية ويحتاج وقتاً طويلاً.



كيف يعمل التعلم المعزز؟

يعمل التعلم المعزز عن طريق برمجة خوارزميات ذات هدف مميز ومجموعة محددة من القواعد لتحقيق هذا الهدف. حيث يتم برمجة الخوارزمية للحصول على مكافآت إيجابية عند تحقيق أمر مفيد متعلق بالهدف النهائي. وتتلقى العقوبات عند اتخاذ إجراء بعيد عن الهدف المطلوب تحقيقه. وعادة ما يستخدم هذه التعلم في:





تطبيقات عمليّة في الذكاء الاصطناعي (تعلّم الآلة)

هذا نموذج بيانات لشركة توصيل تحت الطلب:

اليوم	التاريخ	الوقت	المتصل	من الموقع	إلى الموقع
الأحد	2023-02-20	8:00	33225852	المنامة	مدينة عيسى
الأحد	2023-02-20	14:00	33225852	مدينة عيسى	المنامة
الأحد	2023-05-21	8:30	39323232	البديع	الرفاع
الأحد	2023-05-21	8:30	33225852	المنامة	مدينة عيسى
الاثنين	2023-05-12	12:00	34345656	مدينة عيسى	المنامة
الثلاثاء	2023-05-30	16:00	32324444	المنامة	مدينة عيسى

السؤال: كيف يمكن الاستفادة من هذه البيانات لتحسين أرباح الشركة وتقليل التكاليف ومكافأة الزبائن الأوفياء؟



1- أولاً يجب تحديد المدخلات والمخرجات المطلوبة.

a. ماهي المخرجات التي يمكن الاستفادة منها؟:

i. موقع الطلب: سيمكنني من توفير السيارات المطلوبة في ذلك الموقع عوض توفيرها في مواقع غير مطلوبة.

ii. زمن الطلب: يحدّد لي زمن وجوب توفر السيارات في الموقع الأكثر طلباً في هذا التوقيت.

iii. المتصل: يمكنني من تحديد الزبائن الأكثر ولاءاً للشركة.

b. المدخلات: كل البيانات المتوفرة في الجدول.

2- تحديد طريقة تعلّم الآلة واختيار اللوغاريتم المناسب لاستخراج المطلوب.

a. ما هو نوع اللوغاريتم المناسب في هذه الحالة: هو التعلّم الآلي الخاضع للإشراف: حيث من خلال إدخال كمّ كافٍ من

البيانات وتصنيفها يمكن الاستدلال من خلالها بأكثر الأوقات وأكثر المواقع طلباً للخدمة.

المصادر:

[1] IBM.com – What is Machine Learning

[2] TechnologyReview.com – What is Machine Learning

[3]Expert.ai – What is the Definition of Machine Learning