مُقدِّمة في لغات البرمجة

لغة البرمجة (Programming Language)

قبل التعرف على مفهوم لغة لبرمجة ، هل سألت نفسك يوما ما هي بالبرمجة؟

يمكن تعريف البرمجة بانها عملية كتابة تعليمات وأوامر يمكن للحاسوب فهمها وتنفيذها. إذ تُعد البرمجة الوسيلة التي تتيح للبشر التواصل مع الحواسيب والأجهزة الذكية، حيث يتم تحويل الأفكار إلى برامج وتطبيقات تُنفذ مهام محددة. وحيث أن الحاسوب لا يفهم إلا لغة الآلة (Machine Language)التي تتكون من (0,1)، كان لا بدّ من وجود لغات تُبسط هذا التواصل بين لغة الإنسان ولغة الآلة . وهذا ما يُطلق عليه اسم لغة البرمجة.

اذن يمكن يمكننا القول بأن:

لغة البرمجة : هي مجموعة من الاوامر والتعليمات التي تستخدم في كتابة البرامج والتطبيقات ضمن قواعد محددة.وهي الاداة الرئيسة التي يستخدمها المبرمجون لغايات التواصل والتفاعل مع جهاز الحاسوب لغايات تمكينه من القيام بمهام محددة.

تصنيفاتها:

يمكن تصنيف لغات البرمجة اعتمادًا على عدة معايير منها:

- الوظيفة المناطة بلغة البرمجة
- التطبيقات التي تعتمدها لغة البرمجة
- الآلية والطرق التي تعالجها لغة البرمجة
- درجة قرب لغة البرمجة من اللغات الإنسانية.

والتي على أساسها يمكن تصنيف لغات البرمجة الى نوعين:

ويمكن تصنيف لغات البرمجة الى نوعين آخرين حسب طريقة تمثيل أجزاء البرنامج:

إضاءةٌ :

برنامج الحاسوب هو مجموعة من الأوامر التي تُكتَب بإحدى لغات البرمجة؛ بهدف حَلِّ مشكلة ما، أو أداء مهمة مُحدَّدة باستخدام جهاز الحاسوب

وفيما يلي شرح تفصيلي حول كل نوع من التصنيفات السابقة:

أوَّلًا :لغات البرمجة مُنخفِضة المستوى (Low-Level Languages)

هي نوع من لغات البرمجة التي لا تحتاج إلى مترجم أو مُجَمّع لترجمة التعليمات البرمجية إلى مجموعة من تعليمات الجهاز، كونها أقرب إلى لغة الآلة وبالتالي يصعب على الإنسان فهمها. وتعتبر ضرورية لعمل أنظمة الكمبيوتر.

وتنقسم الى قسمين:

1.لغة الآلة(Machine Language)

هي اللغة الرقمية الأساسية التي يمكن للحاسوب فهمها وتنفيذ التعليمات من خلالها،حيث تتم قراءتها بواسطة وحدة المعالجة المركزية(CPU) ، حيث تتم كتابة التعليمات باستخدام النظام الأساسي الثنائي، حيث يتم تمثيل كل تعليمة بسلسلة من الأرقام الأصفار (0) والواحدات(1). وهذه اللغة يصعب على الإنسان فهمها.

2.لغة التجميع(Assembly Language)

هي مجموعة من التعليمات بحيث كل تعليمة تضم 32 بتاً (خانة) ، وكل تعليمة بلغة التجميع تقابلها ترجمة لها في لغة الآلة وذلك حتى تستطيع وحدة المعالجة المركزية CPU فهمها وتنفيذها،و تقوم هذه اللغة على استخدام برنامج خاص يُسمّى المجمع (Assembler) ويعمل على تحويل الأوامر المكتوبة إلى لغة الآلة التي يفهمها جهاز الحاسوب. تمتاز هذه اللغة بأنَّها أسهل من لغة الآلة؛ نظرًا إلى احتوائها على بعض مفردات اللغة ا لإنجليزية؛
ما يُسهِّل قراءة برامجها وفهمها .غير أنَّ تنفيذ البرامج المكتوبة بلغة التجميع يكون أبطأ مقارنةً بلغة الآلة.

ثانيًا :لغات البرمجة عالية المستوى (High-Level Languages)

هي صنف من أصناف لغات البرمجة التي لا تتطلب من المبرمج الدخول في تفاصيل قوانين تطبيق أوامر البرمجة، إذ
تمتاز لغات البرمجة عالية المستوى بمواءمتها
للغات التي يفهمها الإنسان،حيث يستخدم مفردات
برمجة (syntax) وأوامر (instructions) قريبة من اللغة التي يستخدمها البشر في الكلام.

وسميت بهذا الإسم كونها بعيدة عن لغة الآلة وقريبة من لغة الإنسان. فهي تلائم جميع أنواع أجهزة الحاسوب بغض النظر عن أنظمة تشغيلها ونوعيتها.

ومن أشهر الأمثلة على هذه اللغات:

لغة بايثون Python ، ولغة جافا Java ، ولغة (C++) ولغة (C#).

لغات البرمجة الكتلية (Block-Based Programming (Languages

إنّ بيئات البرمجة المبنية على الكتل أصبحت هي التجربة البرمجية الأولى للعديد من الطلاب لما تمتاز به من سهولة القراءة. حيث أظهرت الأبحاث أن لغات البرمجة المبنية على الكتل كانت فعالة في تدريس مفاهيم علوم الكمبيوتر مثل الخوارزميات والمتغيرات والمنطق الشرطي وفهم الكود. حيث يتم استخدام الكتل الرسومية (**Graphical Blocks**) بدلاً من استخدام النصوص.

ومن أشهر لغات البرمجة الكتلية سكراتش (**Scratch**).

لغات البرمجة النصية (Text-Based Programming (Languages

لغات تُستخدَم فيها النصوص
لتمثيل أجزاء البرامج بدلًا من الكتل الرسومية،إذ تعتبر مجموعة من التعليمات المكتوبة في شكل نص مما يمكن الفريق من تفسيرها وتنفيذها.

حيث تعتبر ملائمة جدًا للمطورين ومسؤولي النظام حيث يمكنهم إجراء التغييرات ورؤية النتائج على الفور.
ومن أشهر لغات البرمجة النصية لغة جافا سكريبت

نشاط(أُقارِن وأُناقِش): أُقارِن بين لغات البرمجة المختلفة، ثمَّ أُناقِش زملائي/ زميلاتي في الكيفية التي تتغيَّر فيها طبيعة البرمجة تبعًا لتغيُّر المزايا في كل لغة برمجة.

لغات ال	لغات البرمجة عالية المستوى	لغات البرمجة الكتلية	لغات البرمجة النصية
أقرب إلى • تسهل وت توفر مكتبات ا • الكود لل • أمثل	• أقرب إلى لغة الآلة. تمنح المبرمج تحكمًا دقيقًا في الأجهزة. يصعب على الانسان فهمها • الكود يكون أكثر تعقيدًا وصعب القراءة. • أمثلة: لغة التجميع(Assembly)	تُستخدَم فيها الكتل الرسومية Graphical Blocks لتمثيل أجزاء البرامج بدلًا من النصوص، مثل لغة البرمجة سكراتش Scratch	ا النصوص لتمثيل أجزاء البرامج من الكتل الرسومية، مثل لغة جافا سكريبت Java Script

الكيفية التي تتغيَّر فيها طبيعة البرمجة تبعًا لتغيُّر المزايا في

- كلما ارتفع مستوى اللغة، قلت الحاجة إلى التفاصيل التقنية، لانها توفر مكتبات وأدوات جاهزة لتسريع التطوير
- تختلف اللغات في سرعة تنفيذ البرامج، حيث تكون اللغات منخفضة المستوى أ
- اللغات التي تشجع على كتابة كود منظم تسهل عملية الصيانة والتطوير عل

المُترجِم والمُفسِّر (Compiler and Interpreter)

لا بدّ وان سبق لك وتابعت يوما مقطع فيديو بإحدى اللغات الأجنبية (لغة المصدر)مصاحباً له ترجمة مكتوبة أو مسموعة (لغة الهدف).او

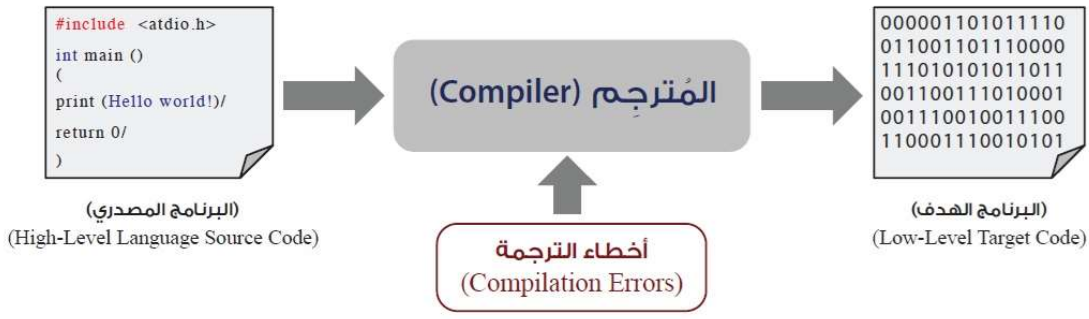
استمعت لخطاب احدى الشخصيات الاجنبية تصاحبه ترجمة فورية مسموعة.
إن الشخص الذي يقوم بتحويل لغة الشخصية الأساسية الى لغة المستمع او المشاهد يطلق عليه اسم المترجم.

وكما سبق وتعلمنا،فإن جهاز الكمبيوتر يفهم لغة الآلة ولا يفهم لغة البشر. من هنا كان لا بد من وجود وسيط بين المبرمج وجهاز الحاسوب يطلق عليه اسم المترجم (**Compiler**).
إن المفسر والمترجم كلاهما أدوات لتحويل الكود البرمجي إلى تعليمات يفهمها الحاسوب، لكنهما يختلفان في طريقة التنفيذ.

اعتمادًا على ما سبق، يمكننا تعريف المترجم كما يلي:

المترجم (Compiler) هو :

برنامج حاسوبي مساعد، يقوم بتحويل البيانات المكتوبة بلغة برمجية عالية المستوى (البرنامج المصدري)إلى مجموعة من تعليمات لغة الآلة منخفضة المستوى القابلة للتنفيذ (البرنامج الهدف)، والتي يمكن فهمها بواسطة وحدة المعالجة المركزية للحاسوب، وذلك دون تغيير معنى الكود.
ويمثل الرسم التالي رسما توضيحيا لما يقوم به المترجم أثناء التعامل مع البرامج المكتوبه باللغات عالية المستوى:



المفسر (Interpreter)

المفسر هو أحد مكونات بيئة تطوير البرمجيات، وظيفته تنفيذ الكود البرمجي مباشرة دون الحاجة إلى ترجمته بالكامل وتحويله الى ملف تنفيذي(Execution File) كما يفعل المترجم، بحيث يقوم المفسر على قراءة الكود في (البرنامج المصدري) سطرًا بعد سطر، وتحويله إلى تعليمات يمكن للمعالج فهمها وتنفيذها فورًا. وفي حال وجود خطأ بأحد الأسطر،يتوقف المفسر عن عملية تحويل الأجزاء المتبقية من البرنامج المصدري.

المفسر والمترجم كلاهما أدوات لتحويل الكود البرمجي إلى تعليمات يفهمها الحاسوب، لكنهما يختلفان في طريقة التنفيذ ،ويمثل الجدول التالي الفرق بينهما:

المترجم(Compiler)	المفسر(Interpreter)
يترجم الكود بالكامل مرة واحدة إلى ملف تنفيذي مستقل مثل ملف (exe)	ينفذ الكود مباشرة سطرًا بسطر أثناء وقت التشغيل.
بعد الترجمة، يتم تنفيذ البرنامج بسرعة عالية دون الحاجة إلى التفسير مرة أخرى.	لا ينتج ملفًا تنفيذيًا مستقلًا.
عملية الترجمة قد تستغرق وقتًا لكنها تعطي أداء أفضل في التشغيل النهائي.	بطيء نسبيًا لأن التفسير يتم في كل مرة يتم فيها تشغيل البرنامج.
ستخدم في لغات مثل C و.C++	مناسب للتطوير والاختبار السريع ويستخدم في لغات مثل Python , PHP

أبحث وأُقارِن: أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أمثلة على كلٍّ من المُترجِم والمُفسِّر، ثمَّ أُقارِن بينهما من حيث آليَّة التنفيذ، والسرعة في التنفيذ، وسهولة اكتشاف الأخطاء.

أمثلة على لغات برمجة تستخدم المترجم والمفسر:

- لغات تستخدم المترجم بشكل أساسي: C، C++، #C
- لغات تستخدم المفسر بشكل أساسي: Python ،JavaScript

من حيث	لغات تستخدم المترجم	لغات تستخدم المفسر
السرعة في التنفيذ	عالية	منخفضة
سهولة اكتشاف الأخطاء.	بعد الترجمة	أثناء التنفيذ
آليَّة التنفيذ	يقوم المترجم بالفحص الكامل لأيِّ برنامج كُتِب بلغة البرمجة عالية المستوى (البرنامج المصدري)، ثمَّ ترجمته إلى لغة الآلة (البرنامج الهدف)	يقوم المفسر بتحويل كل جزء من أجزاء البرنامج المكتوب بلغة البرمجة عالية المستوى إلى لغة الآلة، ثمَّ تنفيذ هذه الأجزاء أمرًا بأمر

الخوارزميات (Algorithms) :

يمكن تعريف الخوارزمية على انها
مجموعةٌ منَ التعليماتِ أوِ الخطواتِ المنطقيَّةِ والمنظَّمةِ التي تُحدِّدُ كيفيَّةَ حلِّ م شكلةٍ معيَّنةٍ أوْ إنجازِ مهمةٍ محدَّدةٍ. وتعتبر جزءاً أساسياً من عالم البرمجة الحاسوبية،حيث يتم استخدامها للقيام بعمليات عديدة مثل البحث في قواعد البيانات أو إجراء الحسابات.

سميت الخوارزمية بهذا الاسم نسبة إلى عالم الرياضيات المسلم أبو جعفر محمد بن موسى الخوارزمي الذي ابتكرها في القرن التاسع الميلادي.

وبما انها تعمل على حل مشكلة،إذن لا بدّ من وجود معطيات لهذه المشكلة ومن ثم إجراءات للعمل على حلها، وأخيرا ظهور الحل لهذه المشكلة.

إنّ معطيات المشكلة تمثل ما يسمى **بالمدخلات**

وإجراءات العمل لحلها تمثل **مجموعة القواعد والتعليمات**

وظهور الحل يمثل **المخرجات**

وعليه يمكننا تمثيل مفهوم الخوارزمية على الشكل التالي:

يراعى عند كتابة الخوارزميات مجموعة من المعايير والضوابط، أبرزها:

1. التسلسل المنطقي للخطوات والتعليمات.
2. التحديد الواضح والدقيق للمدخلات التي تدخل الخوارزمية، والمخرجات التي تنتج منها.
3. وضوح الخطوات، وسهولة تتبُّعها.
4. الفاعلية مُمثَّلةً في سرعة تنفيذها.

طرق تمثيل الخوارزمية:

يقصد بعملية تمثيل الخوارزمية الطريقة التي يمكن من خلالها التعبير عن الخوارزمية ويمكن تعريفه بأنه عملية تحويل الخوارزمية من شكلها المجرد الى شكل يمكن فهمه وتنفيذه بسهولة من قبل الآخرين .حيث أن استخدام طريقة منظمة للتعبير عن الخوارزميات يتيح حرية التعبير عن الحل للمشكلة

مع الاحتفاظ بإمكانية فهم الحل من قبل أي شخص وسهولة تحويله الى لغة برمجة من قبل المبرمجين والمطورين.

ويمكن تمثيل الخوارزمية بطريقتين كما بالشكل التالي:

1Pseudo code.شبه الرمزية(شبيه الكود)

ويطلق عليها الكود الزائف،حيث تعتبر طريقة للتعبير عن الخوارزمية من خلال لغة وسيطة تشبه لغة البرمجة في طريقة كتابتها إلا انها لا تلتزم بقواعد البرمجة المرتبطة باللغة والتي يجب اتباعها عند كتابة كود البرمجة.ولكن يسهل فهمها من قبل المبرمجين ومن لهم معرفة بلغات البرمجة. حيث تُكتَب الخوارزمية شبه الرمزية في مجموعة من الخطوات المُرقَّمة، تُستخدَم فيها لغة الإنسان، والتعابير والرموز الرياضية ال بسيطة.

مثال: (1)

أكتب خوارزمية شبه رمزية لإيجاد ناتج المعادلة : y = a * x + b، ثمَّ طباعتها.

الحَلُّ :أتَّبِع الخطوات الآتية في الحَلِّ:

1. أبدأ.
2. أُدخِل قِيَم المُتغيِّرات a :، و x، و b.
3. أَجِد ناتج ضرب a * x
4. أَجِد قيمة y بتطبيق المعادلة الآتية y = a * x + b
5. أطبع قيمة y.
6. أتوقَّف.

2. مُخطَّطات سَيْر العمليات (Flowcharts):

ويطلق عليها أيضا الخرائط الإنسيابية او خارطة الإنسياب أو المخطط التدفق وهي عبارة عن طريقة تمثيل رسومي لخطوات الخوارزمية من خلال استخدام الأشكال الهندسية مُتعارَف عليها، ومجموعة من الأسهم والخطوط التي تُحدِّد سَيْر الخوارزمية.

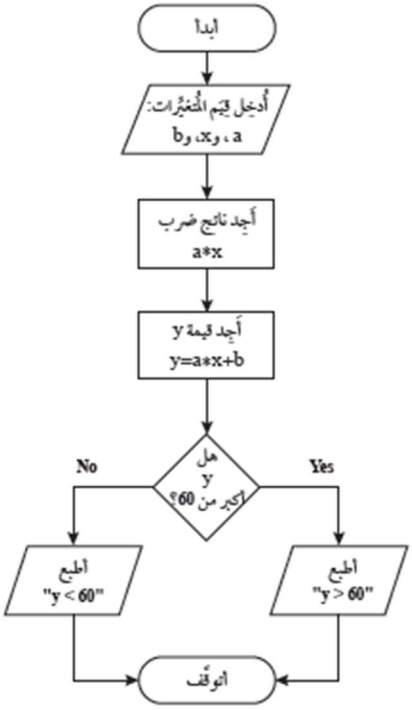
وكل شكل من هذه الأشكال الظاهرة في الجدول يدلُّ على خطوة مُعيَّنة من خطوات تمثيل الخوارزمية المتفق عليها:

اسمُ الشكلِ	الشكلُ	الوصفُ
نقطةُ البدايةِ أوِ النهايةِ		تمثلُ نقطةَ البدايةِ أوِ النهايةِ لبرنامجٍ ما.
عمليةٌ أوْ إجراءٌ		يمثلُ مهمةً أوْ إجـــراءً محددًا يُنفَّذُ، مثلَ الحساباتِ أوْ معالجةِ البياناتِ.
إدخالٌ/ إخراجٌ		يمثلُ إدخالَ أوْ إخراجَ البياناتِ منْ وإلى البرنامجِ، مثلَ قراءةِ إدخالِ المستخدمِ أوْ عرضِ النتائجِ.
القرارُ أوِ الشروطُ		يمثلُ نقطةَ تفرعٍ في البرنامجِ حيثُ يُقيَّمُ شرطٌ، وتُتَّخذُ إجراءاتٌ مختلفةٌ بناءً على النتيجةِ، وعادةً ما تمثَّلُ كشـــكلٍ معينٍ. تحتوي عادةً على عبارةٍ شـــرطيةٍ، مثلَ عباراتِ if، أوْ حلقاتِ التكرارِ.
السهمُ		تُشيرُ الأسهمُ إلى تدفّقِ الرسمِ البيانيِّ.

مثال:

لنعد الى المثال السابق ونقوم بالتعديل عليه بحيث تطبع الخوارزمية **(60 < y)** إذا كان الناتج أكبر من 60،
وتطبع **(y < 60)** إذا كان الناتج أقل من 60، إضافةً إلى طباعة الناتج **y**

الحل



المواطنةُ الرقميةُ:

- أحرص دائمًا على استخدام البرمجيات المُرخَّصة قانونيًّا، واحترام حقوق المُلْكية الفكرية للمُطوّرين والشركات.
- أتجنَّب استخدام البرمجيات المُقرصَنة أو البرمجيات غير المُرخَّصة؛ لأنَّها قد تكون غير آمنة، وتُعرِّض جهازي ومعلوماتي الشخصية للخطر.