الفصل الرابع
اختيار بيئة البرمجة

تعنى بيئة البرمجة الأدوات التى يستخدمها المبرمج للبرمجة ، وقد تكون هذه الأدوات لغـــة بـــرمجة ، أو مجموعة لغات برمجة ، أو مجموعة من الأدوات ، وطريقة حل مشاكل البرمجة .

تنقســم بيـــئات الـــبرمجة إلى بيئة متكاملة ، وبيئة غير متكاملة ، بيئة البرمجة الغير متكاملة هى أدوات مستقلة ، أما بيئة البرمجة المتكاملة فهى دمج لأدوات البرمجة فى بيـــئة واحـــدة مثل لغة فيجوال بيزك التى تحتوى على محرر لكتابة البرامج وأدوات وأزرار قوائـــم وشـــرائط أدوات ومفسر ومترجم فى بيئة واحدة ، وتعتمد على طرق لحل مشاكل البرمجة .

يعتبر اختيار بيئة برمجة مناسبة عملية تقديرية صعبة إذ تعتبر بيئات البرمجة متنافسة فيمـــا بينها ، فبعضها يوفر تقنيات وأدوات سهلة الاستخدام ، وبعضها يوفر إمكانيات كبيرة كما يوفر بعضها استخدام طرق معينة لحل مشاكل البرمجة .

يتوقف اختيار بيئة البرمجة المناسبة على : الخلفية السابقة للمبرمج ، ونوعية الجهاز ، ونظـــام التشـــغيل الذى يعمل عليه ، أو الذى يرغب المبرمج فى تصميم البرنامج عليه ، ومدى توافر لغة البرمجة فى المؤسسة التى يعمل عليها ، ونوعية البرامج التى يقوم المبرمج بتصميمها والغرض منها .

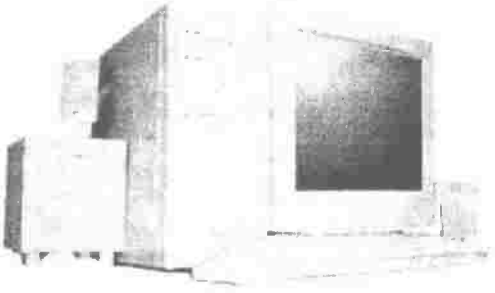
يتوقف اختيار بيئة البرمجة فى أغلب الأحوال على منصة الكمبيوتر Platform ، أو النظام الذى تم تصميم الكمبيوتر من أجله ، وتتكون منصة الكمبيوتر من :

- المكونات المادية (العتاد) Hardware للكمبيوتر .
- ونظام التشغيل Operating System الذى يعمل عليه جهاز الكمبيوتر .

العتاد والبرامج

يعد عتاد الحاسب Hardware المكون الأساسى للجهاز الذى يحدد نظام البرمجة فكل جهاز يتكون من مكونات مادية تشتمل على :

- صـــندوق النظام الذى يحتوى على : الدوائر الإلكترونية ، والمعالج ، والذاكرة ، ووحدات التحكم مع الأقراص والبطاقات المختلفة .
- الشاشة .
- لوحة المفاتيح .
- الفأرة .
- أى أجهزة أخرى داخلية أو خارجية مثل : المودم ، وبطاقات الصوت والفيديو ، ومحـــركات الأقراص المضغوطة والمرنة ، والأقراص الصلبة ، والطابعة ، وغيرها من الملحقات الأخرى للإدخال والإخراج .



لهـــذه المكونـــات تأثيـــر عـــلى البرمجة فعلى سبيل المثال يؤثر نوع المعالج الدقيق Microprocessor (وحـــدة المعالجـــة المركـــزية CPU) على استخدام لغة التجميع Assembly ، الـــتى تعتمد على نوع المعالج ، إذ يختلف برنامج لغة تجميع مكتوب لمعالج بنتيوم Pentium عن برنامج آخر مكتوب لمعالج موتورلا Motorola .

كمـــثال آخر فإن البرنامج المعد للاستخدام مع جهاز إنتاج شركة آى بى ام IBM أو متوافق معه Compatible يختلف عن برنامج آخر مكتوب لجهاز أبل ماكنتوش .

كمثال ثالث فإن البرنامج المكتوب ليستخدم الفأرة Mouse فى جهاز إنتاج شركة آى بى ام IBM ، أو مـــتوافق معه Compatible يعمل باستخدام فأرة ذات زرين ، وهو بـــرنامج يختلف عن برنامج آخر مكتوب لجهاز أبل ماكنتوش لفأرة قياسية ذات زر واحد .

كمـــثال رابع لتأثير نوعية ومكونات الجهاز على البرمجة قد تنتج من تصميم برنامج يعمـــل باستخدام لوحة مفاتيح للإدخال ، أو استخدام الإدخال عن طريق شاشة لمس ،

أو باستخدام قلم إدخال ، أو لوح .

لما كان البرنامج يحدد مصدر الإدخال (لوحة مفاتيح أو شاشة لمس أو ماسحا ضوئيا أو عصا لعب أو فأرة) ، فإنه أيضا يكون مصمما ليعالج البيانات بمعالج دقيق معين له إمكانيات معينة ، ويتولى توجيه الحاسب لإخراج مخرجاته على وحدة الإخراج المناسبة (شاشة أو طابعة أو قرص) ، وبهذا فإن وظائف البرنامج الأساسية (الإدخال والمعالجة والإخراج) تعتمد اعتمادا أساسيا على المكونات المادية للكمبيوتر .

من الواضح أنه يجب فهم النظام المادى للجهاز الذى يعمل عليه البرنامج ليمكن تصميم وإنشاء البرنامج الذى يعمل على هذه المكونات المادية .

على الجانب الآخر من مكونات منصة النظام Platform فإن الجانب الثانى بعد المكونات المادية هو نظام التشغيل Operating System الذى يؤثر على البرمجة .

نظام التشغيل والبرامج

تشير كلمة برامج Software إلى أى برنامج (مجموعة تعليمات) يوجه الكمبيوتر لينفذ مهمة أو وظيفة ما ، ويمكن تقسيم البرمجيات فى فئتين هما :

* أنظمة التشغيل Operating Systems .
* التطبيقات Applications .

نظام التشغيل هو مجموعة برمجية خاصة يتم تحميلها عند بدء تشغيل الكمبيوتر من قرص النظام الذى يعمل عليه الجهاز ، ويكون نظام التشغيل مسئولا عن تشغيل الكمبيوتر بالإضافة إلى إدارة كل العتاد المادة والموارد (الذاكرة والمعالجة والطباعة وغيرها من موارد النظام) ، ويوفر واجهة بين عتاد الكمبيوتر والمستخدم .

تستخدم أنظمة التشغيل لإدارة العتاد Hardware ، والبيانات Data ، والتطبيقات Applications ، لذلك فهى أساسية فى عمل الكمبيوتر ، ومن أمثلتها نظام تشغيل القرص دوس MS DOS ، وواجهة ويندوز Windows 3.x ، ونظام تشغيل ويندوز بإصداراته المختلفة Windows 95, 98, Windows NT, Windows 2000, Windows XP ، ونظام تشغيل OS/2 ، ونظام تشغيل يونيكس Unix ، ونظام تشغيل System لأجهزة أبل .

اعتمد المطورون بشكل أساسى على برمجيات نظم التشغيل الأولى (دوس - ويندوز

3.x) قبل الوصول إلى أنظمة تشغيل 32 بت الحديثة مثل نظام تشغيل ويندوز بإصداراته المختلفة حتى أمكن الوصول إلى أنظمة تشغيل 64 بت فى إصدار ويندوز Windows XP 64 Bit .

التطبيقات هى الأدوات التى يوظفها المستخدم ، وتستخدم التطبيقات (مثل وورد وأكسس واكسل وغيرها) نظام التشغيل لتجعل المستخدم قادرا على إنشاء ومعالجة وعرض البيانات .

لما كان نظام التشغيل هو (مجموعة برامج تدير المكونات المادية للكمبيوتر) فإن كل جهاز كمبيوتر مهما كانت نوعيات مكوناته المادية يحتوى على نظام تشغيل .

تختلف نظم تشغيل الحاسبات الشخصية PC عن نظم تشغيل الحاسبات الكبيرة والمتوسطة ، وأيضا تختلف أنظمة تشغيل حاسبات آى بى ام عن أنظمة تشغيل حاسبات ماكنتوش .

تعتمد أجهزة حاسبات آى بى ام والمتوافقة معها على نظم تشغيل مايكروسوفت فى الغالبية العظمى منها : (أنظمة دوس DOS وويندوز WINDOWS بإصدارات مختلفة) ، وتوجد أنظمة تشغيل أخرى غير شائعة الاستخدام للحاسب الشخصى مثل نظام لينوكس ، لكن كل نظم التشغيل تقوم بأداء نفس المهام ، وتختلف فى طريقة أداء هذه المهام .

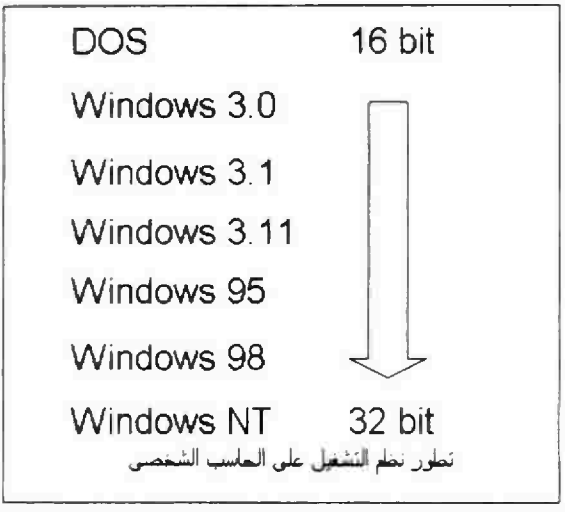
نظرا لقيام نظام التشغيل بالتحكم فى عناصر المكونات المادية المختلفة فإن المبرمج يستخدم وظائفية نظام التشغيل فى الإدخال والإخراج والمعالجة ، كما يستفيد من شفرة نظام التشغيل فى برامجه التى يقوم بكتابتها .

من بين أمثلة الوظائف التى يستفيد منها المبرمج فى نظام التشغيل : نظم إدارة الملفات File Management ، والتخزين Storage على الأقراص ، والطباعة Print ، والاتصالات ، ولما كان نظام التشغيل يقوم بإعداد هذه الوظائف فإنها تصبح متاحة للمبرمج فى صورة وظائف يمكنه الوصول إليها واستدعائها .

تقوم لغة البرمجة المستخدمة بالاستفادة من (واجهة برمجة التطبيقات Application Programming Interface) فى نظام التشغيل ، وهى مجموعة الوظائف التى يمكن الوصول إليها فى نظام التشغيل .

حروف دوس DOS اختصار كلمات نظام تشغيل القرص Disk Operating System ، وقد صدرت إصدارات لنظام تشغيل دوس Version آخرها دوس ٧ مع ويندوز Windows 98 .

لا تعتبر نوافذ مايكروسوفت Windows 3.x نظام تشغيل مستقل ، وإنما تعتبر واجهة تشغيل تستخدم نظام دوس كأساس لها ، فى حين أن ويندوز Windows 95 ، وما أعقبه من إصدارات ويندوز ، ونظام ويندوز ان تى Windows NT بإصداراته المختلفة هى نظم تشغيل مستقلة عن نظام دوس ، وليست مجرد طبقة إدارة ذاكرة وواجهة مستخدم رسومية .



تطور نظم التشغيل على الحاسب الشخصى

كان الهدف من نظام ويندوز : إيجاد واجهة مستخدم رسومية ، تظهر الأوامر فيها على شكل صور أو رموز بدلا من إدخالها فى سطر الأوامر كما فى نظام دوس ، وتم تصميم نظام ويندوز لتستخدم الفأرة بشكل أساسى كجهاز تأشير ، كما يمكن استخدام لوحة المفاتيح ، وتوفر جميع تطبيقات معايير ويندوز واجهة مستخدم رسومية GUI Graphical User Interface تتضمن الآتى :

١- نظام قائمة Menu بالأوامر الأساسية .

٢- يتم تحديد النص أو الكائنات بأسلوب متشابه .

٣- وظائف نقر وسحب الفأرة واحدة لجميع التطبيقات .

بفضل هذه الوظائف المشتركة أصبح المستخدم قادرا على تعلم التطبيقات بسرعة ،

كمـــا يســـتطيع مطـــور التطبيقات أيضا الوصول إلى مجموعة أدوات من برمجيات ويندوز ، مما يوفر إعادة كتابة الأشياء نفسها فى برمجة جديدة .

يستخدم ويندوز نظاما معقدا لإدارة الذاكرة يجعله قادرا على الاستفادة بشكل أمثل من الذاكرة ، وبالتالى يتيح للمستخدم تشغيل عدة تطبيقات فى الوقت نفسه ، والتنقل بينها للقيام بأعمال التحرير مثل : النسخ Copy ، والقص Cut ، واللصق Paste .

نظـــام ويندوز هو نظام يدير الموارد ، أى أنه يتعامل مع كل شئ فى الكمبيوتر على أنـــه مـــورد ، وتضـــم المـــوارد : موارد Resources الذاكرة ، والعرض المرئى ، والمـــنافذ ، وبطاقة الصوت ، ويتم تقديم جميع الموارد إلى ويندوز من خلال سواقات الجهـــاز التى هى : ملفات تعرف كيفية التخاطب مع ويندوز ، أى أنها تتيح التخاطب بين الجهاز ونظام ويندوز ، لهذا تجد أن نظام واجهة ويندوز هو مدير موارد .

تظهـــر أهمية تعريف الموارد فى ويندوز عند كتابة تطبيقات تستخدمها ، إذ يمكن فى هـــذه الحالـــة طلب وصول ويندوز إلى هذه الموارد بدلا من كتابة شفرة برنامج يقوم بهذا (كان مطور التطبيقات فى بيئة دوس يقوم بعمل جميع هذه المهام بنفسه) .

تطورت البرمجيات وأنظمة التشغيل ، وعلى الرغم من أن نظام تشغيل دوس محدود بقيـــود حجم الذاكرة ، وسهولة التكوين ، فقد جاء نظام ويندوز الذى يعمل بالاعتماد عـــلى نظـــام دوس ليحـــل هذه المشاكل إلى حد ما ، ثم جاء نظام ويندوز 95 كنظام تشـــغيل كـــامل مستقل ، وتبعه نظام ويندوز 98 لمتابعة احتياجات التكنولوجيا ونمو شـــبكة الإنـــترنت ، ثم ظهر ويندوز 2000 ليستجيب لدواعى تطور الوسائط المتعددة والتشبيك المشترك ، وبعده جاء نظام ويندوز XP كآخر إصدارات ويندوز (كان نظام ويـــندوز ان تى Windows NT قد ظهر بأقوى من احتياجات الكمبيوتر الشخصى ، ومصمما أساسا للعمل الشبكى) .

إن هذا التطوير نفسه يفرض على المبرمج الاستفادة من التقنيات الإضافية ، والأدوات التى يتيحها نظام التشغيل ، ويفرض هذا على المبرمج فهم نظام التشغيل ، وإمكاناته ، وطـــريقة أداء وظائفـــه ، فكتابة برنامج لنظام يونكس يختلف عن كتابة برنامج لنظام ويندوز كما يختلف أيضا عن كتابة برنامج لنظام دوس أو غيره من نظم التشغيل .

اختيار نوع البرمجة

تطـورت بيـئات البرمجة وطرقها تطورا كبيرا ، وتعددت الحلول المختلفة لمشاكل الـبرمجة ، وعند إنشاء برنامج خاص يتعرض المبرمج للعديد من المشاكل التى يريد إيجاد حل لها ، وتختلف طريقة الحل التى تستخدم لحل مشاكل البرمجة اعتمادا على الطريقة المتبعة لهذا الحل من بين الطرق التالية :

- البرمجة التركيبية .
- أو برمجة الوحدات .
- أو البرمجة النمطية .

توفر هذه الطرق ما يلى :

أ- التخلص من تكرار السطور .

ب- تسهيل قراءة البرامج ، فبرنامج مقسم إلى أجزاء أسهل من برنامج كبير .

ج- تبسـيط تطويـر الـبرنامج ، فتقسيم البرنامج يسهل التصميم ، والكتابة ، وإزالة الأخطاء والعلل ، كما يمكن فريق العمل من تبادل الوحدات .

د- يمكن إعادة استعمالها فى برامج أخرى .

هـ- توسيع لغة البرمجة .

البرمجة التركيبية

يسـتخدم فى هذا النوع من البرمجة تركيبات للتحكم فى ترتيب أداء أوامر البرمجة ، وتـتيح معظـم لغـات البرمجة استخدام الأوامر التى توفر إنشاء البرامج بسهولة ، ويستفيد هذا النوع من تحقق الشروط والتكرار .

برمجة الوحدات

تـتيح لغـات البرمجة إنشاء وحدات نمطية ، أو بمعنى آخر تتيح تقسيم البرنامج إلى مكونـات جزئية تقوم كل منها بتنفيذ مجموعة من الأوامر يمكن تنفيذ كل منها بشكل مسـتقل عن الأخرى ، وقد تتداخل مع بعضها البعض (بمعنى أداء عملية محددة أثناء تنفيذ عملية أخرى) .

تـتيح برمجة الوحدات التعامل مع أجزاء صغيرة من شفرة البرنامج بدلا من التعامل

مـــع أجزاء كبيرة ، وبالتالى يمكن تصحيح الأخطاء بسهولة ، وتتبع منطق البرنامج بيســـر ، كمـــا تفيد عند عمل فريق عمل فى برنامج واحد يكون كل فرد فى الفريق مسئولا عن جزء محدد صغير لمهمة معينة .

البرمجة النمطية

تشـــبه بـــرمجة الأنمـــاط أو ما يطلق عليه اسم البرمجة الشيئية أو البرمجة غرضية الـــتوجه Object Oriented Programming (OOP) بـــرمجة الوحدات فى كثير من الوجـــوه ، إذ يتم إنشاء وحدات نمطية تقوم بتنفيذ مهمة محددة لكن يمكن استخدام تلك الوحدات النمطية مرات أخرى عن طريق نقلها إلى برامج أخرى .

إن فائدة استخدام البرمجة النمطية يتعدى مجرد إعادة استخدامها إلى سهولة تصحيحها وسهولة تعديلها ، فبينما تكون برامج البرمجة التقليدية كبيرة بحيث يصعب تتبعها فإن الـــبرمجة التقـــليدية تعانى أيضا من أنه يتم الاحتفاظ بالبيانات المستخدمة فى برنامج بشـــكل منفصل عن شفرة البرنامج ، إن هذا لا يعنى فقط صعوبة التعديل لكنه أيضا يسبب استهلاكا للذاكرة ، وتدنى سرعة البرنامج ، إذ يتم الاحتفاظ بالبيانات فى منطقة واحدة داخل ذاكرة الكمبيوتر ، ويتم تخزين شفرة البرنامج فى منطقة أخرى .

حلت البرمجة النمطية هذه المشاكل فلم يعد البرنامج كبيرا بل أصبحت مكوناته أجزاء صغيرة يمكن تعديلها بسهولة ، كما يتم تخزين البيانات مع الوحدات النمطية للبرنامج ممـــا يعنى أن تقدر الأجزاء المختلفة من البرنامج على الوصول إلى الجزء المسئول وطلب البيانات ، مما يوفر سرعة وتكاملا ، وسهولة فى تعديل البيانات بتعديل الوحدة المسئولة إضافة إلى التنظيم الجيد لمنطق وهيكل البرنامج .

اختيار لغة البرمجة

بسبب العدد الكبير من لغات البرمجة المستخدمة فى مجال الكمبيوتر ، ووجود العديد مـــن الإصـــدارات للغة الواحدة ، إضافة إلى طبيعة الاحتياجات التى قد يحتاج إليها المبرمج أو البرنامج نفسه ، فإن تحديد لغة البرمجة المناسبة هو أكثر عناصر امتلاك موهبة البرمجة تعقيدا .

لا توجـــد لغـــة مناسبة ، ولغة غير مناسبة لكن توجد احتياجات وإمكانيات ، وهناك

العديد من العوامل التى تؤثر فى تحديد لغة البرمجة التى تناسب الاحتياجات ، وتلبى إمكاناتها المتطلبات ، ويمكن من الحصول عليها واستخدامها ، ومن هذه العوامل :

١- الإتاحة والإمكانية .

٢- عامل التوافقية .

٣- الاختيار الخارجى .

٤- العيوب والميزات .

٥- منصة الكمبيوتر .

فيما يلى توضيح هذه العوامل :

١- الإتاحة والإمكانية

عند بداية تقدير الاحتياجات المناسبة من أدوات ولغات البرمجة فإن أول ما يتبادر إلى ذهن المبرمج هو : المتيسر من اللغات ، وإمكانية الحصول عليها ، فتكلفة لغات البرامج عالية ، لكن نظم التشغيل قد تتضمن على الأقل واحدة من لغات البرمجة (مثل نسخة من لغة بيزك Qbasic المتوفرة مع نظام ويندوز بإصدارته المختلفة ، ونظام دوس) .

تتوفر على شبكة الإنترنت مواقع توفر الحصول على نسخ تجريبية من لغات البرمجة ، كما توفر بعض الشركات نسخا مجانية من لغات البرمجة التى تصدرها .

قد يتاح الحصول على نسخة من نسخ لغات البرمجة المتوفرة فى مؤسسة من المؤسسات التى تعمل بها .

الحقيقة أنه يمكن الحصول على لغة ما من لغات البرمجة ، لكن الحصول على المتيسر والمتاح لا يعنى أن هذه اللغة قد تلبى متطلبات البرمجة التى يسعى إليها المبرمج ، وتقدير مدى ملاءمة اللغة للاحتياجات .

قد يكون استخدام اللغة المتاحة مناسبا للتعلم أو للأغراض التدريبية لكنه أيضا قد يوجه التدريب والتعلم نحو لغة برمجة غير ملائمة للاستخدام المطلوب ، إلا أن هذا الأمر لا ينفى أن استخدام المتاح والمتيسر سوف يكون أساسا قويا لتعلم وفهم أسس البرمجة ، وطرق استخدام اللغة ، ومعرفة قواعد الاستخدام ، وتعلم أساليب وطرق لغات البرمجة فى كتابة البرامج .

٢- عامل التوافقية

يعد عامل التوافقية Compatibility من أهم عوامل اختيار لغة البرمجة ، ليس فقط بسبب إمكانية استخدام برامج قديمة قام بها المبرمج ، ولا يحتاج فقط سوى تغيير أجزاء من شفرة البرنامج لتناسب الإصدار الجديد ، أو إجراء تعديلات طفيفة على البرنامج ، لكن أيضا بسبب خبرة المبرمج فى هذه اللغة إضافة إلى سمة توافق الإصدار الجديد من البرنامج مع الإصدار الجديد لنفس البرنامج الذى قام بعمله المبرمج .

٣- الاختيار الخارجى

قد لا يستطيع المبرمج اختيار اللغة بناء على اختياراته أو إمكانياته ، بل قد تفرض عليه لغة برمجة معينة لا يملك اختيارها ، عندما يذهب إلى مؤسسة من المؤسسات التى تملك أو تصنع برامجها بلغة برمجة معينة ، وتحتاج إلى التوافق مع برامجها القديمة ، أو استخدام بياناتها القديمة ، أو نتيجة سياسة معينة للمؤسسة التى يلتحق بها المبرمج ، لذلك قد تكون هناك لغة برمجة تخضع للاختيار الخارجى .

٤- العيوب والميزات

يخضع اختيار لغة برمجة معينة أيضا لميزات وعيوب لغة البرمجة ، فلكل لغة ميزات تتفوق بها على اللغات الأخرى ، كما أن لكل لغة برمجة العديد من المثالب والعيوب ، فهناك لغات برمجة قوية فى التعامل مع قواعد البيانات والملفات ، وهناك لغات برمجة قوية إزاء الرسوم والأصوات والوسائط المتعددة ، وهناك لغات برمجة تمتاز بمعالجة المعادلات الرياضية ، وعلى الجانب الآخر هناك لغات برمجة تعانى ضعفا فى معالجة النصوص ، أو غيرها تعانى ضعفا فى تداول ومعالجة البيانات والملفات ، بينما تعانى بعض لغات البرمجة من قصور فى معالجة الوسائط المتعددة . المعنى الواضح من وراء ذلك هو : أنه لا توجد لغة برمجة يمكنها تنفيذ كل مهام البرمجة ، وقد يحتاج الأمر إلى استخدام أكثر من لغة برمجة ، وفى هذه الحالة سيكون من الضرورى فهم عيوب ومزايا كل منها ، إضافة إلى فهم طريقة الربط بين اللغتين ، أو استخدام الأجزاء البرمجية بين كل من اللغتين وتداولها بينهما .

قد لا يكون المبرمج المبتدئ على دراية كاملة بمزايا وعيوب لغات البرمجة المختلفة لكنه يستطيع السؤال والاستفسار والاطلاع للبحث عن المزايا التى يتطلع إلى استخدامها لتتكون لديه خلفية عن اللغات ومزايا وعيوب كل منها .

٥- منصة الكمبيوتر

يخضع اختيار لغة برمجة معينة أيضا لمنصة جهاز الكمبيوتر نفسه وقدرته على تشغيل هذه اللغة .

البرامج الأولى ولغة البرمجة الأولى

إن كتابة البرامج ليس فقط سهلا وإنما يمكن أن يصبح ممتعا أيضا ، وتسرى جميع مفاهيم البرمجة على أى لغة أخرى تتعلمها فى المستقبل ، وتدعم خطوات المستقبل فى عالم البرمجة ، وتشمل النقاط الأساسية :

تثبيت اللغة - اكتشاف خصائص اللغة - المحرر واستخدامه - عناصر اللغة وكلماتها المحجوزة - المتغيرات وأهمية أسمائها - الفرق بين المتغيرات النصية والرقمية - كيفية إجراء العمليات الحسابية - رؤية نتائج العمليات الحسابية - التكرار ، والتفرع ، والشروط - كيفية إدخال برنامج وتنفيذه - ماذا تفعل عند ظهور أخطاء - كيفية طباعة النتائج - معرفة كيفية تخزين البيانات - كتابة البرامج .

لابد من استخدام محرر لكتابة البرامج ، والمحرر هو شاشة إدخال على هيئة معالج نصوص يمكن من خلاله كتابة شفرة البرنامج ، وقد يكون المحرر أكثر من مجرد محرر : فقد يكون مفسرا مثل Interpreter QBasic ، ومن خلال المحرر يمكن كتابة البرنامج وتشغيله وتعديله وعرض مخرجاته وحفظ البرنامج على القرص .

على الرغم من اختلاف جميع لغات البرمجة فأثناء دراسة لغة ما ستلاحظ أن جميع اللغات تقريبا لديها نقاط مشتركة مع بعضها البعض فهى لديها : طرق لإدخال البيانات ، وطرق لطباعة المخرجات ، واستخراج القيم من لوحة المفاتيح والقرص ، وتحتوى لغات البرمجة على أوامر لتنفيذ الحلقات التكرارية Loop ، كما يمكنها اتخاذ القرارات ، وتقوم بعض اللغات بأداء هذه الإجراءات بتلقائية ومهارة أكثر من لغات أخرى .

الأهم أن لغة البرمجة الثانية التى تتعلمها تكون دائما أسهل من الأولى .

عند تعلم كتابة البرامج من المهم أن تكون سهلة التعديل والصيانة ، فالبرامج التى تقوم بكتابتها لا يكفى أن تعمل فقط ، بل يجب أن تكون سهلة القراءة ومفهومة ، وذلك بغرض تحديثها فى المستقبل فوجود كود واضح يجعل هذه المهمة أسهل بكثير .

هناك ميزة أخرى فى كتابة البرامج بشكل منظم وواضح ، وهى : الانتهاء من كتابة البرامج بسرعة ، وقلة عدد الأخطاء .

هناك بعض العبارات الهامة التى يمكن وضعها فى برنامج عبارة عن ملحوظات تصف ما يؤديه البرنامج ، ويحتوى أى برنامج ملحوظات يتجاهلها مفسر أو مترجم اللغة ، وتصف الملحوظات العمل الذى سيؤديه البرنامج بلغة سهلة وبسيطة .

إن عملية معالجة البيانات تعنى أن يقوم البرنامج بتشغيل البيانات التى يجب أن تكون مخزنة بطريقة ما فى الذاكرة ، وذلك أثناء قيام برنامجك بالتعامل معها وتتماثل أغلب لغات البرمجة فى تخزين البيانات فى متغيرات (يمكن التفكير فى المتغير كصندوق يحتوى على قيمة البيانات داخل الكمبيوتر وهذه القيمة يمكن أن تكون رقما أو حرفا أو مجموعة من الحروف) .

يمكن أن تستوعب البرامج أى متغيرات ، وهناك حد أقصى لعدد المتغيرات التى يستوعبها برنامج واحد فى لغة ما ، ويعتمد هذا الحد على نوع البيانات التى يحتويها هذا البرنامج ، وهو كاف على أية حال ، ويصاحب المتغيرات أسماء مرتبطة بها .

إن المدخلات والمخرجات هى حجر أساس برنامج يتفاعل مع العالم الخارجى ، ويمكن إنتاج مخرجات على الشاشة أو الطابعة ، وتنسيق شكل المخرجات لتحسين مظهر الطباعة وتحديد التباعد بين قيم المخرجات المتكررة على نفس السطر .

تستخدم جمل الإدخال المختلفة لإدخال البيانات من لوحة المفاتيح أو نقر الفأرة أو عند حدوث حدث ما أو من القرص ، ويمكن بواسطة خيارات هذه الأوامر تنسيق شكل المدخلات وتحديد نوعيتها ، وإن إتقان شكل المدخلات والمخرجات يحدث الفرق بين برنامج جيد وآخر أقل جودة ، ويوضح ما يفضله المستخدم .