Task1

المقدمة:

- الاجابة عن الاسئلة الاتية:
- ما هي البرمجة؟
- ما هو الكود البرمجي؟
- أنواع لغات البرمجة(High-level vs Low-level)
- الفرق بين الفرونت إند والباك إند ما هو الـFramework ؟
- ما هوAPI ؟
- ما هو(Version Control (Git ؟
- ما هو الـ IDE وText Editor ؟
- ما هوDebugging ؟
- الفرق بين Compiler و Interpreter
- ما هو agile

ما هي البرمجة؟

لغة البرمجة هي نظام ترميز أو وسيلة تستخدم لإعطاء التعليمات للحاسوب أو أي جهاز آخر للقيام بأعمال محددة. هي مجموعة من القواعد والرموز والكلمات التي تستخدم لكتابة البرامج التي تمكن الحاسوب من أداء المهام المختلفة.

ما هو الكود البرمجي؟

تعريف الكود البرمجي هو نظام إشارات يستخدم لتمثيل الحروف أو الأرقام في إرسال رسائل التعليمات الموجودة في برنامج الكمبيوتر، وغالبا ما تسمى التعليمات التي يكتبها المبرمج بلغة برمجة التعليمات البرمجية المصدر، كما تسمى التعليمات التي تم تحويلها إلى لغة الآلة التي يفهمها الكمبيوتر رمز الجهاز أو التعليمات البرمجية القابلة للتنفيذ.

أنواع لغات البرمجة(High-level vs Low-level)

تتعدد أنواع لغات البرمجة اعتمادًا على عدة عوامل، من بينها مستوى الترميز وطريقة التنفيذ والغرض من الاستخدام. إليك بعض أنواع لغات البرمجة الرئيسية .1 :حسب مستوى الترميز :لغات منخفضة المستوى :مثل لغة الآلة ولغات التجميع، وهي قريبة من بنية الحاسوب مباشرة وتوفر أداءً عاليًا .لغات عالية المستوى :مثل سي، سي++، جافا، بايثون، وهي أكثر سهولة في الاستخدام وتوفر أدوات وأساليب برمجة متقدمة.

لغات البرمجة تصنف حسب مستوى الترميز إلى لغات منخفضة المستوى (مثل لغة الآلة ولغة التجميع) ولغات عالية المستوى (مثل بايثون وجافا وسي++). لغات منخفضة المستوى تكون قريبة من لغة الجهاز وتتطلب فهمًا دقيقًا لبنية الحاسوب، بينما لغات عالية المستوى تكون أسهل للفهم والقراءة وتوفر أدوات مثل المترجمين والمفسرين لتسهيل عملية البرمجة .توضيح إضافي :لغات منخفضة المستوى :(Low-level languages) لغة الآلة :(Machine language) هي اللغة التي يفهمها الحاسوب مباشرة، وهي عبارة عن سلسلة من الأرقام الثنائية (0 و 1 .(لغة التجميع (Assembly language):هي لغة برمجة بسيطة تُمثل تعليمات الجهاز بشكل رمزي، وهي أكثر سهولة من لغة الآلة ولكنها لا تزال تتطلب معرفة جيدة ببنية الحاسوب .لغات عالية المستوى :(High-level languages) هي لغات برمجة قريبة من اللغة البشرية، مثل بايثون وجافا وسي .++توفر أدوات مثل المترجمين والمفسرين التي تترجم الكود البرمجي إلى لغة الآلة، مما يسهل عملية البرمجة .تتميز بسهولة قراءتها وتركيزها على حل المشكلات بدلاً من تفاصيل الجهاز .أمثلة على لغات البرمجة :منخفضة المستوى: لغة الآلة، لغة التجميع .عالية المستوى: بايثون، جافا، سي++، سي شارب.

الفرق بين الفرونت إند والباك إند ما هو الـFramework ؟

الواجهة الأمامية (Frontend) هي الجزء الذي يمكن لمستخدمي موقع الويب رؤيته والتفاعل معه مثل واجهة المستخدم الرسومية (GUI) بما في ذلك التصميم والتنقل في القوائم والنصوص والصور ومقاطع الفيديو وما إلى ذلك. على العكس من ذلك، فإن الواجهة الخلفية (Backend) هي الجزء الذي لا يستطيع مستخدمو الموقع رؤيته والتفاعل معه.

ما هوAPI ؟

واجهة برمجة التطبيقات (API) هي مجموعة من القواعد أو البروتوكولات التي تمكن تطبيقين أو أكثر من التواصل مع بعضهما البعض لتبادل البيانات أو الميزات أو الوظائف. تُعد API بمثابة وسيط أو واجهة يتيح للتطبيقات المختلفة العمل معًا بدلاً من أن يكون كل تطبيق منفصلاً .أمثلة على استخدام واجهات برمجة التطبيقات :تطبيقات التواصل الاجتماعي : تستخدم واجهات API لتسجيل الدخول إلى حسابك، وعرض المشاركات، والنشر عليها .تطبيقات الملاحة :تستخدم واجهات APIللحصول على بيانات الخرائط وبيانات حركة المرور .تطبيقات الألعاب :تستخدم واجهات API لتسجيل الدخول، وتخزين سجلات اللاعبين، وتوفير ميزات متعددة اللاعبين .تطبيقات التجارة الإلكترونية :تستخدم واجهات API لإجراء معاملات مالية، والحصول على معلومات المنتج، وإدارة المخزون .كيف تعمل واجهة برمجة التطبيقات؟ إرسال الطلب : يقوم تطبيق ما بإرسال طلب إلى واجهة برمجة التطبيقات، محددًا العملية التي يريدها (مثل "احصل على بيانات المنتج ."(الاستجابة :تستجيب واجهة برمجة التطبيقات ببيانات أو معلومات، بناءً على الطلب الذي تم إرساله .التواصل :يتمكن التطبيق الآخر من استخدامه هذه البيانات أو المعلومات في تطبيقاته .أهمية واجهات برمجة التطبيقات :تبسيط عملية التطوير :تسهل واجهات برمجة التطبيقات عملية تطوير التطبيقات من خلال توفير وظائف جاهزة للاستخدام .زيادة كفاءة التطبيقات :تتيح واجهات برمجة التطبيقات للتطبيقات التواصل مع بعضها البعض بشكل أكثر كفاءة .توفير أمان أكبر : تسمح واجهات برمجة التطبيقات بمشاركة المعلومات الضرورية فقط، مع إبقاء تفاصيل النظام الداخلية الأخرى مخفية . توسيع نطاق التطبيقات :تتيح واجهات برمجة التطبيقات للتطبيقات الوصول إلى وظائف وخدمات أخرى، مما يوسع نطاق التطبيقات .ما هي واجهة برمجة التطبيقات(API) ؟ AWS - مترجَم — ماذا تعنيAPI ؟ API هي اختصار لعبارة "واجهة

برمجة التطبيقات". في سياق واجهات برمجة التطبيقات، يشير مصطلح "تطبيق" إلى أي برنام aws.amazon.com ...
ما هي واجهة برمجة التطبيقات(API) ؟ IBM - مترجَم — ما هوAPI ؟ واجهة برمجة التطبيقات (API) هي مجموعة من القواعد أو البروتوكولات التي تمكن تطبيقات البرامج من التواصل مع بعضها ال ibm.com ...ما المقصود بواجهة API؟ Apple Support - تُستخدمAPI ، وهي اختصار لواجهة برمجة التطبيقات، لتمرير البيانات ذهابًا وإيابًا بين تطبيقات البرامج بطريقة رسمية. توفر العديد من الخدم..

ما هو(Git) Version Control ؟

ما هو الـ IDE وText Editor ؟

محرر النصوص (Text Editor) هو برنامج لتعديل ملفات نصية، بينما بيئة التطوير المتكاملة (IDE) هي برنامج أكثر تعقيدًا يجمع بين محرر نصوص ومصحح أخطاء ومترجم وأدوات أخرى لتطوير البرمجيات .محرر النصوص (Text Editor):الوظيفة: يستخدم أساسًا لكتابة وتحرير ملفات نصية بسيطة، مثل النصوص، التعليمات البرمجية، أو ملفات التكوين .الميزات: يتيح كتابة النص، التنسيق، وحفظه. قد يتضمن بعض الميزات مثل تمييز بناء الجملة (syntax highlighting)لمساعدة المبرمجين في قراءة الكود بسهولة .أمثلة: Notepad++، Sublime Text، Atom.بيئة التطوير المتكاملة (IDE): الوظيفة: هي برنامج متكامل يجمع بين عدة أدوات لتطوير البرمجيات، بما في ذلك محرر نصوص، ومصحح أخطاء، ومترجم، وادوات أخرى مثل أدوات التحكم في الإصدار، وأدوات بناء البرامج، وأدوات الاختبار .الميزات: يوفر بيئة شاملة لعملية تطوير البرمجيات، بدءًا من كتابة الكود حتى تنفيذه واختباره .أمثلة: Visual Studio، Eclipse، IntelliJ IDEA.الفرق الأساسي :محرر النصوص هو أداة أساسية لكتابة النصوص .بيئة التطوير المتكاملة هي أداة متكاملة لتطوير البرمجيات، وتضم ميزات متقدمة لمساعدة المبرمجين في تطوير البرامج بسهولة وكفاءة .الاستخدام :محرر النصوص مناسب للمهام التي لا تحتاج إلى أدوات إضافية، مثل تحرير النصوص البسيطة، أو كتابة التعليمات البرمجية الصغيرة .بيئة التطوير المتكاملة مفضلة للمبرمجين الذين يعملون على مشاريع أكبر وأكثر تعقيدًا، حيث توفر أدوات متكاملة لتطوير البرامج، مثل مصحح الأخطاء والمترجم

ما هوDebugging ؟

Debugging:تصحيح الأخطاء هو عملية البحث عن أي خطأ أو خلل وإصلاحه في رمز المصدر لأي برنامج. عندما لا يعمل البرنامج كما هو متوقع، يدرس مبرمجو الكمبيوتر الرمز لتحديد سبب حدوث أي أخطاء. ويستخدمون أدوات تصحيح الأخطاء لتشغيل البرنامج في بيئة خاضعة للرقابة، والتحقق من الرمز بالتفصيل، وتحليل المشكلة وحلها.

الفرق بين Compiler و Interpreter

الفرق الأساسي بين Compiler و Interpreter هو الطريقة التي يتعامل بها مع الكود المصدر. المترجم (Compiler) يترجم الكود بأكمله إلى لغة الآلة قبل التنفيذ، بينما المفسر (Interpreter) يترجم الكود سطرًا بسطر أثناء التنفيذ .تفاصيل أكثر :المترجم (Compiler): يقرأ الكود المصدر بالكامل .يُترجم الكود إلى لغة الآلة أو كود وسطي قابل للتنفيذ .ينتج ملفًا قابلًا للتنفيذ .يمكنه اكتشاف الأخطاء بعد تحليل الكود بالكامل .أمثلة لغات البرمجة التي تستخدم Compiler: C, C++, Java.المفسر (Interpreter): يقرأ الكود المصدر سطرًا بسطر .يُترجم الكود ويُنفذه في نفس الوقت .لا ينتج ملفًا قابلًا للتنفيذ بشكل مستقل .يُمكنه تتبع الأخطاء مباشرةً أثناء التنفيذ .أمثلة لغات البرمجة التي تستخدم Interpreter: Python, PHP, Ruby.

ما هو agile

منهجية Agile هي إطار عمل لإدارة المشاريع يقسم المشاريع إلى عدة مراحل ديناميكية، تُعرف عادةً باسم العدو السريع . إطار العمل الرشيق هو منهجية تكرارية . بعد كل سباق، تتأمل الفرق وتنظر إلى الوراء لترى ما إذا كان هناك ما يمكن تحسينه لتعديل استراتيجيتها للسباق التالي.

Agileهي منهجية تطوير تعتمد على نهج تكراري وتدريجي .Scrum هو تطبيق لمنهجيةAgile ، حيث تُجرى التغييرات التدريجية في الوقت المناسب.

ما الفرق بين Agile وScrum ؟

Agileهي منهجية تطوير تعتمد على نهج تكراري وتدريجي.

Scrumهو تطبيق لمنهجيةAgile ، حيث تُجرى التغييرات التدريجية في الوقت المناسب.

ما هو(Git) Version Control ؟

أنظمة التحكم في الإصدارات (VCSs) هي برامج تتتبع التغييرات في الكود المصدري، أو البرمجة الأساسية للبرنامج.
عند بناء مشاريع برمجية واسعة النطاق، تتغير الأمور كثيرًا وبسرعة - فقد يقرر العملاء إضافة ميزة مختلفة، أو قد يوجه مدير المشروع جزءًا من المشروع في اتجاه مختلف (ليغير رأيه بعد بضعة أسابيع). مع أنظمة التحكم في الإصدارات مثلGit ، يمكن للمهندسين الرجوع إلى الإصدارات القديمة من الكود والتراجع عن التغييرات، مما يضمن استمرار المشاريع في التطور والتكيف بسلاسة، ويحافظ على الكود نظيفًا