Qué es un IDE o Entorno de Desarrollo Integrado

Un IDE, del inglés Integrated Development Environment, es un Entorno de Desarrollo Integrado, que te permite maximizar tu productividad como programador, proporcionando facilidades para el desarrollo de software.

Qué es un IDE para programación

Un IDE (Entorno de Desarrollo Integrado) es una herramienta de software que proporciona un entorno de programación completo para los desarrolladores de software. Este conjunto de herramientas es utilizado para ayudar al desarrollo de software desde un mismo techo.

Los programadores las emplean para escribir, probar y depurar sus programas de acuerdo con normas precisas. Para esto, incluye diferentes herramientas como editores de texto, depuradores o interfaces visuales adaptadas a varios lenguajes de programación.

En su esencia más básica, un IDE es un editor de texto. Aquí tienes algunos ejemplos de archivos que puedes editar con un IDE:

.txt

.json

.js

.ts

.java

.kt

.md

.html

También puede ser una aplicación independiente que sirve de entorno para otras herramientas basadas en la línea de comandos, como compiladores e intérpretes.

Características y Componentes de un IDE

Un IDE usualmente está compuesto por:

Un editor de código

Herramientas de automatización

Depurador o debugger

La mayoría de los IDEs modernos ahora cuentan con un completador de código inteligente. El caso más sonado de esta funcionalidad es el nuevo Github Copilot, que usa inteligencia artificial para ayudarte a terminar tu código.

Adicionalmente, algunos IDE contienen un compilador, un intérprete o ambos. El límite entre un entorno de desarrollo integrado y otras partes del entorno de desarrollo de software más amplio no está claramente definido. A veces se integra un sistema de control de versiones (como GIT), o varias herramientas para simplificar la construcción de una interfaz gráfica de usuario (GUI).

que\_son\_ide.jpg

Cuando estás estudiando lenguajes de programación y desarrollando software, es importante comenzar por elegir tu editor de texto y entender qué es un IDE. Esto es algo que te quedará muy claro al momento de tomar nuestro curso Gratis de programación básica.

La gran mayoría de los editores de texto tienen funciones que facilitan la escritura y edición a niveles avanzados. De hecho, una de las cosas más simples que hacen es resaltar diferentes partes de tu código con colores para que puedas reconocer y diferenciar strings, métodos o clases. Algunos IDEs también incluyen navegadores de clases, objetos y diagramas de jerarquía de clases para cuando haces programación orientada a objetos (POO).

Ejemplos de IDE más populares

Algunos IDE más populares entre los programadores son:

Visual Studio Code

Sublime Text

Atom

Vim

Eclipse

Android Studio

NetBeans

JetBrains IntelliJ

Apple Xcode

Komodo

Acá puedes ver un ejemplo real utilizando el IDE Atom:

Para qué sirve un IDE en desarrollo de software

El IDE está construido para ayudarte en el desarrollo de software duante las siguientes fases:

Escribir código

Compilar

Configurar

Hacer ‘Debug’ o testing

Por lo general, el IDE contiene la estructura y las convenciones de estilo de varios lenguajes de programación para que tú puedas programar en cualquier lenguaje y recibir alertas y recomendaciones sobre tu código, logrando con esto que sea más limpio y legible.

Funciones adicionales de los entornos de desarrollo

Algunas de las características típicas de los IDE incluyen herramientas de depuración, llenado automático de ciertos elementos de código y la capacidad de detectar errores a medida que programas.

Para proyectos que abarcan varios archivos, el IDE examina los archivos y ayuda a mantener la coherencia en todo el proyecto. Los IDE pueden facilitar las pruebas de código e identificar partes de tu proyecto que puedes refactorizar (“pimpear” tu código para que sea elegante y sencillo de leer).

Ojo: el IDE no hace magia ni programa por ti. Es fundamental que tú mantengas un orden e integres un sistema de control de versiones. Esto es fundamental para poder tener éxito como programador.

Qué IDEs son mejores para programar en Python

Si te gusta desarrollar en Python (como a mí), te recomiendo utilizar alguno de los IDEs más populares para este lenguaje:

PyCharm

PyDev

Spyder

Si aún no estás familiarizado con Python o quieres aprender este lenguaje, te recomiendo que tomes este curso:

Curso Básico de Python

Sin duda es uno de mis cursos favoritos en Platzi. Échale un ojo y cuéntame en los comentarios qué más quisieras aprender.

¿Cuàl es la diferencia entre programaciòn y interactividad?

programacion es codigo que tu escribes para que nuestros telefonos o computadores hagan algo.

la interactividad es el resultado de la programaciòn.

html = no tiene interactividad

css = no tiene interactividad

javascript = si tiene interactividad con la informacion de html

Html = es un lenguaje de marcado de hipertexto y hay colocamos la informaciòn.

css = Es un lenguaje de estilo en cascado, el contenido lo muestra de una forma visual.

javascrpit = hace que los botones funcionen.

¿con que extensiones se guarda?

html = .html

css = .css

javascript = .js

¿Que editores de codigo hay?

sublime text

atom = editor del curso

visual studio

¿Interpreta los archivos?

es el que habre = navegador

Clase numero 13

Tarea investiga sobre do while

The do...while statement creates a loop that executes a specified statement until the test condition evaluates to false. The condition is evaluated after executing the statement, resulting in the specified statement executing at least once.

statement

A statement that is executed at least once and is re-executed each time the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ /\* ... \*/ }) to group those statements.

condition

An expression evaluated after each pass through the loop. If condition evaluates to true, the statement is re-executed. When condition evaluates to false, control passes to the statement following the do...while.

Note: Use the break statement to stop a loop before condition evaluates to true.

.

function dibujarTeclado(evento)

{

if(evento.keyCode == teclas.UP)

{

console.log("Vamo' pa arriba");

}

if(evento.keyCode == teclas.DOWN)

{

console.log("Vamo' pa abajo");

}

if(evento.keyCode == teclas.LEFT)

{

console.log("Vamo' pa izquierda");

Para variables aleatorias

var z;

for(var i=0; i<10; i++)

{

z = aleatorio(-20, 20);

document.write(z + ", ");

}

var cantidad = aleatorio

var z;

for(var i=0; i<10; i++)

{

z = aleatorio(-20, 20);

document.write(z + ", ");

}