Kotlin Fundamentals

Reactivos

Pregunta 1

¿Qué sistema de automatización de construcción de software utiliza Kotlin?

Opciones:

a. Gradel
b. Maven
c. Jenkins
d. Docker

Pregunta 2

En las variables de entorno del sistema, ¿cuál es el propósito de configurar la variable JAVA_HOME?

Opciones:

a. Indicar la ubicación de los juegos Java instalados.
b. Establecer la ruta del JDK instalado.
c. Indicar la versión de Java en uso.
d. Configurar el servidor Java.

Pregunta 3

Al configurar el entorno de desarrollo para Kotlin, ¿qué debes verificar después de configurar las variables de entorno para el JDK?

Opciones:

a. Que el navegador web reconozca Java.
b. Que el IDE muestre un mensaje de bienvenida.
c. Que el comando javac -version muestre la versión del JDK instalado.
d. Que el sistema operativo se reinicie automáticamente.

Pregunta 4

Si estás trabajando en un proyecto de Kotlin y decides utilizar IntelliJ IDEA como tu IDE, ¿qué debes hacer si te encuentras con un error relacionado con la versión del JDK?

Opciones:

a. Cambiar a otro IDE.
b. Ignorar el error y continuar programando.
c. Actualizar a la versión más reciente del JDK.
d. Bajar la versión del JDK.

Pregunta 5

Si estás iniciando un nuevo proyecto en Kotlin usando IntelliJ IDEA, ¿qué herramienta de construcción deberías configurar según el documento?

Opciones:

a. Gradel
b. Maven
c. Jenkins
d. Docker

Pregunta 6

En el IDE IntelliJ Idea, ¿qué representa la carpeta src en un proyecto Kotlin?

Opciones:

a. Contiene las librerías de automatización de compilación.
b. Es el *source* del proyecto y almacena los recursos que se utilizarán.
c. Es la carpeta raíz del proyecto.
d. Es donde se almacenan las configuraciones del IDE.

Pregunta 7

¿Cómo se declara una variable en Kotlin?

Opciones:

a. Usando la palabra clave const.
b. Usando la palabra clave let.
c. Usando la palabra clave var.
d. Usando la palabra clave val para cualquier tipo de variable.

Pregunta 8

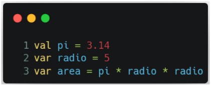
En Kotlin, ¿cuál es el tamaño en bits de un tipo de dato Int?

Opciones:

a. 8 bits
b. 16 bits
c. 64 bits
d. 32 bits

Pregunta 9

En el siguiente código Kotlin, ¿cuál es el valor de la variable "área" después de ejecutarlo?



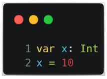
```
val pi = 3.14
var radio = 5
var area = pi * radio * radio
```

Opciones:

a. 15.7
b. 31.4
c. 78.5
d. Ninguna de las anteriores

Pregunta 10

En el siguiente código en Kotlin, ¿qué tipo de dato tiene la variable x?



```
var x: Int
x = 10
```

Opciones:

a. Double
b. Float
c. Long
d. Int

Pregunta 11

En programación, ¿qué son las estructuras condicionales?

Opciones:

a. Son herramientas que permiten repetir bloques de código.
b. Son elementos que permiten condicionar o repetir una acción o grupo de acciones.
c. Son herramientas exclusivas de Kotlin.
d. Son estructuras que solo trabajan con números.

Pregunta 12

En Kotlin, ¿cuál de los siguientes operadores se utiliza para verificar la igualdad?

Opciones:

a. ==
b. =
c. !=
d. <=

Pregunta 13

En Kotlin, ¿cuál es el equivalente al switch de Java?

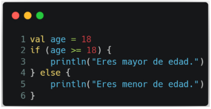
Opciones:

a. If

b. When
c. Else
d. Switch

Pregunta 14

De acuerdo con el siguiente código en Kotlin, ¿qué se imprimirá en la consola?



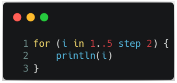
```
val age = 18
if (age >= 18) {
    println("Eres mayor de edad.")
} else {
    println("Eres menor de edad.")
}
```

Opciones:

a. Nada
b. Eres menor de edad.
c. Eres mayor de edad.
d. Error de sintaxis.

Pregunta 15

De acuerdo con el siguiente código en Kotlin, ¿cuáles números se imprimirán en la consola?



```
for (i in 1..5 step 2) {
    println(i)
}
```

Opciones:

a. 1, 3, 5
b. 1, 2, 3, 4, 5
c. 2, 4
d. 0, 2, 4

Pregunta 16

En el mundo de la programación y según el documento, ¿cuál es la importancia de las funciones?

Opciones:

a. Son útiles solo en programación orientada a objetos.
b. Son exclusivas de Kotlin.
c. Son opcionales y no esenciales para la programación.
d. Son el componente fundamental para encapsular instrucciones y reutilizar código.

Pregunta 17

En Java y según el documento, ¿qué tipo de funciones son equivalentes a las funciones void?

Opciones:

a. Lambda
b. Funciones de orden superior
c. Unit
d. Funciones de extensión

Pregunta 18

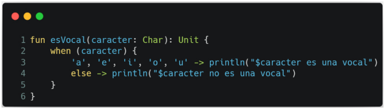
En Kotlin, ¿cómo se define una función?

Opciones:

a. Usando la palabra clave fun.
b. Usando la palabra clave function.
c. Usando la palabra clave def.
d. Usando la palabra clave functionality.

Pregunta 19

De acuerdo con el siguiente código en Kotlin, ¿qué imprimirá si llamamos a la función con esVocal('b')?



```kotlin
fun esVocal(caracter: Char): Unit {
    when (caracter) {
        'a', 'e', 'i', 'o', 'u' -> println("$caracter es una vocal")
        else -> println("$caracter no es una vocal")
    }
}
```

Opciones:

a. “b es una vocal"
b. "b es un número"
c. "b es un carácter especial"
d. Ninguna de las anteriores

Pregunta 20

Si tienes una función en Kotlin que no retorna ningún valor, ¿qué tipo de retorno deberías especificar?

Opciones:

a. Unit
b. Void
c. Null
d. Nothing

Pregunta 21

En Kotlin, ¿qué palabra reservada se utiliza para declarar un método?

Opciones:

a. method
b. fun
c. function
d. procedure

Pregunta 22

En Kotlin, ¿qué es un constructor?

Opciones:

a. Una función que devuelve un valor.
b. Una estructura que define las propiedades de una clase.
c. Se utiliza para inicializar el contenido de las nuevas instancias de una clase.
d. Una herramienta para modificar la visibilidad de una clase.

Pregunta 23

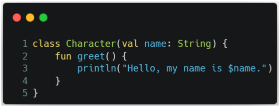
En Kotlin, ¿qué palabra reservada se utiliza para definir una clase?

Opciones:

a. func
b. define
c. object
d. class

Pregunta 24

De acuerdo con el siguiente código en Kotlin: si creas una instancia de la clase Character con el nombre "Alice" y el método se llama greet, ¿qué se imprime?



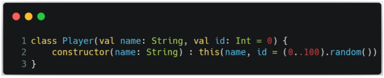
```kotlin
class Character(val name: String) {
    fun greet() {
        println("Hello, my name is $name.")
    }
}
```

Opciones:

a. "Hello, Alice."
b. "Character named Alice."
c. "Name: Alice"
d. "Hello, my name is Alice."

Pregunta 25

De acuerdo con el siguiente código en Kotlin: si creas una instancia de la clase Player solo con el nombre "John", ¿qué sucede con el ID?



```kotlin
class Player(val name: String, val id: Int = 0) {
    constructor(name: String) : this(name, id = (0..100).random())
}
```

Opciones:

a. El ID siempre será 0.
b. El ID será un número aleatorio entre 0 y 100.
c. Se produce un error porque el ID es necesario.
d. El ID será el número de caracteres en el nombre.

Pregunta 26

En Kotlin, ¿qué es la programación orientada a objetos?

Opciones:

a. Es una forma de construir programas donde las entidades principales son funciones.
b. Es una forma de construir programas basados en procedimientos.
c. Es una forma de construir programas donde las entidades principales son los objetos.
d. Es una forma de construir programas basados en eventos.

Pregunta 27

En Kotlin, ¿qué es una clase anidada?

Opciones:

a. Es una clase que se declara dentro de otra clase.
b. Es una clase que se declara fuera de otra clase.
c. Es una clase que se declara dentro de una función.
d. Es una clase que se declara dentro de un paquete.

Pregunta 28

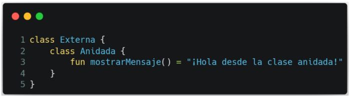
En Kotlin, ¿qué es POJO?

Opciones:

a. Es un patrón de diseño específico para interfaces.
b. Es un enfoque de patrón de diseño de persistencia.
c. Es un tipo especial de función.
d. Es un tipo de variable.

Pregunta 29

De acuerdo con el siguiente código en Kotlin, ¿cómo se crea una instancia de la clase *Anidada*?



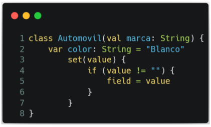
```kotlin
class Externa {
    class Anidada {
        fun mostrarMensaje() = "¡Hola desde la clase anidada!"
    }
}
```

Opciones:

a. Externa().Anidada()
b. Anidada()
c. Externa.Anidada()
d. Externa()

Pregunta 30

De acuerdo con el siguiente código en Kotlin, ¿qué hace el *setter* personalizado en la propiedad *color*?



```kotlin
class Automovil(val marca: String) {
    var color: String = "Blanco"
        set(value) {
            if (value != "") {
                field = value
            }
        }
}
```

Opciones:

a. Establece el color del automóvil como blanco sin importar el valor proporcionado.

b. Establece el color del automóvil solo si el valor proporcionado es "Blanco".
c. No permite cambiar el color del automóvil.
d. Establece el color del automóvil solo si el valor proporcionado no está vacío.

Pregunta 31

¿Cuál es uno de los principales motivos por los que Kotlin está ganando popularidad en el mundo de la programación?

Opciones:

a. Se perfila como el reemplazo definitivo de Java para la plataforma Android.
b. Es más costoso que Java.
c. No es compatible con Android.
d. No puede coexistir con Java en el mismo proyecto.

Pregunta 32

En Android, ¿qué software se requiere para trabajar con Kotlin?

Opciones:

a. Eclipse
b. Android Studio
c. Visual Studio
d. IntelliJ IDEA

Pregunta 33

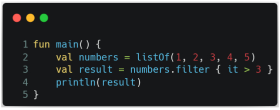
En Android Studio, ¿qué vista no refleja una jerarquía real de los archivos en el disco, sino que los organiza en módulos y tipos de archivos?

Opciones:

a. Vista de Proyecto
b. Vista de Java
c. Vista de Android
d. Vista de Kotlin

Pregunta 34

De acuerdo con el siguiente código en Kotlin, ¿qué se imprimirá en la consola?



```kotlin
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5)
    val result = numbers.filter { it > 3 }
    println(result)
}
```

Opciones:

a. [1, 2, 3]
b. [1, 2, 3, 4, 5]
c. [2, 3, 4]
d. [4, 5]

Pregunta 35

Si deseas desarrollar una aplicación en Kotlin para Android que sea compatible con la mayor cantidad de dispositivos en el mercado, ¿qué deberías considerar al elegir el API Level?

Opciones:

a. Elegir el API Level más reciente.
b. Elegir el API Level más antiguo.
c. Elegir un API Level intermedio basado en las estadísticas de uso.
d. El API Level no afecta la compatibilidad.

Pregunta 36

Es una de las principales características de Kotlin.

Opciones:

a. Es incompatible con Java.
b. No puede ser utilizado para el desarrollo de Android.
c. Tiene su propio entorno de desarrollo.
d. Es compatible con Java y permite codificar partes de un proyecto en ambos lenguajes.

Pregunta 37

En Android, ¿qué es un adaptador?

Opciones:

a. Un tipo de widget.
b. Una clase que representa una lista.
c. Un mecanismo que funciona como un puente entre los datos y las vistas.
d. Una herramienta para la depuración de código.

Pregunta 38

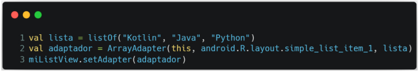
En Android, ¿cuál es la ventaja de usar RecyclerView sobre ListView?

Opciones:

a. RecyclerView no permite desplazamiento.
b. RecyclerView "recicla" los ítems que ya no son visibles por el usuario debido al desplazamiento.
c. ListView es más moderno que RecyclerView.
d. RecyclerView no permite la organización de información.

Pregunta 39

En una aplicación Android, ¿qué realiza este fragmento de código?



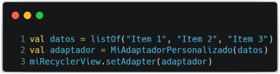
```
val lista = listOf("Kotlin", "Java", "Python")
val adaptador = ArrayAdapter(this, android.R.layout.simple_list_item_1, lista)
miListView.setAdapter(adaptador)
```

Opciones:

a. Muestra una lista de lenguajes de programación en un componente ListView.
b. Crea una base de datos con los nombres de lenguajes de programación.
c. Establece una conexión a una API para recuperar lenguajes de programación.
d. Inicia una nueva actividad que muestra los lenguajes de programación.

Pregunta 40

Si MiAdaptadorPersonalizado es una clase que extiende de RecyclerView.Adapter, ¿qué realiza este fragmento de código?



```
val datos = listOf("Item 1", "Item 2", "Item 3")
val adaptador = MiAdaptadorPersonalizado(datos)
miRecyclerView.setAdapter(adaptador)
```

Opciones:

a. Texto
b. Texto
c. Texto
d. Texto