Fundamentos de Programación

ADIMRA - ITEC

Slides de apoyo para clases sincrónicas
Semana 4

Elementos de lenguaje

Comentarios

Existen elementos genéricos, que podemos observar en todos los lenguajes de programación.

Comenzaremos a repasar los esenciales, entre los que podemos listar:

- Variables y constantes
- Ciclos.
- Sentencias condicionales.
- Operadores (aritméticos, relacionales, lógicos).

Variables y constantes

Una variable no es más que un nombre arbitrario con el cual hacemos referencia a un dato específico dentro de nuestro código fuente.

nro1 = 16

Tal cual lo indica su nombre, este valor puede cambiar, es decir, podemos asignar nuevos valores a esta variable a lo largo del código.

Una constante por su parte, se define como un valor que no se modificará a lo largo de toda la ejecución del código.

Convenciones de nomenclatura

Existen algunas reglas al momento de nombrar los elementos de lenguaje (como variables, contantes, funciones, etc).

Las 4 reglas genéricas son las siguientes:

- Camel Case.
- Pascal Case.
- Snake Case.
- Kebab Case.

Convenciones de nomenclatura

Camel Case:

Cada palabra dentro del nombre comienza con mayúscula, excepto la primera.

Por ejemplo, para una variable que guarde el identificador primario de un producto, escribimos:

idPrimarioProd

Convenciones de nomenclatura

Pascal Case:

Idéntico al anterior, pero la primer palabra también comienza con mayúsculas:

IdPrimarioProd

Convenciones de nomenclatura

Snake Case:

Para identificar más claramente las palabras que componen el nombre del elemento, se utilizan guiones bajos en lugar de mayúsculas:

id_primario_prod

Convenciones de nomenclatura

Kebab Case:

Idem anterior pero empleando guiones medios en lugar de bajos:

id-primario-prod

Estructuras de control. Ciclos.

Un ciclo es un mecanismo que cualquier lenguaje dispone para repetir la ejecución de un determinado paquete de instrucciones.

Es otro de los elementos primarios, ya que esta necesidad es algo muy habitual al codificar, siendo mucho más claro y mantenible el utilizar un ciclo, que simplemente repetir manualmente la misma instrucción una y otra vez:

```
for ciclo in range(10):
    print(ciclo)
```

Estructuras de control. Ciclos.

Generalmente los lenguajes disponen de varias instrucciones relacionadas a ciclos, las más habituales son for, while y do while o do loop por ejemplo.

Realizando una categorización sencilla según al cantidad de iteracciones de un ciclo, tendríamos:

- Infinitos.
- Indefinidos.
- Finitos.

Estructuras de control. Ciclos.

Infinito

Es aquel que queda constantemente iterando, mientras el código principal esté en ejecución.

Si por ejemplo en un sistema, deseamos hacer sonar una alarma intermitente, podemos simplemente dentro de un ciclo infinito:

- activar la alarma.
- aguardar un instante.
- desactivar la alarma.
- aguardar un instante.
- repetir.

Estructuras de control. Ciclos.

Indefinido

Similar al caso anterior, el sistema quedará iterando en este ciclo, pero además estará atento a algún evento externo que pueda modificar su situación.

Siguiendo con el mismo ejemplo de la alarma, si agregamos un pulsador para desactivarla, el ciclo repetirá constantemente la secuencia, pero además quedará atento al estado del pulsador; ni bien este pulsador se active, el ciclo se detendrá.

Estructuras de control. Ciclos.

Finito

Es simplemente un ciclo que itera una x cantidad de veces, como por ejemplo al momento de solicitar determinada cantidad de ingresos al usuario, o recorrer un paquete de datos.

En muchos proyectos de código, aparecerá en algún momento la necesidad de utilizar un ciclo, al menos de alguno de estos tipos, y en oportunidades de 2 o los 3.

Estructuras de control. Condicionales.

Una estructura condicional, es sencillamente un mecanismo para evaluar si una determinada condición (o grupo de condiciones) se verifica o no.

Esto es extremadamente importante, ya que podremos comenzar a decidir y ejecutar diferentes bloques de código, conforme se sucedan distintas situaciones; ya no estaremos limitados a ejecutar la misma secuencia única de pasos.

Estructuras de control. Condicionales.

La instrucción para evaluación de condiciones, presente en todos los lenguajes, es el

if.

Por ejemplo, dado un número, mostrar si es par o impar:

```
numero = 28
if (numero % 2 == 0):
    print("Es PAR")
else:
    print("Es IMPAR")
```

Operadores.

Los operadores son símbolos que nos permiten elaborar operaciones o condiciones.

Los 3 tipos habituales, son:

- Aritméticos.
- Relacionales.
- Lógicos.

Operadores.

Aritméticos:

Son los que estamos más habituados a manejar para **operar**, ya que los vemos desde muy temprana edad al comenzar a aprender matemática en la escuela:

- +: suma.
- -: resta.
- *: multiplicación.
- /: división.
- //: división entera.
- %: módulo (resto de la división).

Operadores.

Relacionales:

Nos permiten **comparar** elementos, son los operadores que empleamos al momento de armar condiciones (if):

- ==: igual a.
- !=: distinto a.
- >: mayor a.
- >=: mayor o igual a.
- <: menor a.
- <=: menor o igual a.

Operadores.

Lógicos (booleanos):

Estos operadores se utilizan habitualmente en conjunción con los relacionales, para **encadenar** diferentes condiciones y evaluarlas como unidad.

- and: y.
- or: o.
- not: no (negación).