

T01: Conceptos básicos y buenas prácticas (1)

1.- Introducción y definiciones básicas

Podríamos definir la programación como la comunicación entre las personas y un ordenador. Se pueden utilizar muchos lenguajes para programar un ordenador. El más básico es el lenguaje de máquina, una colección de instrucciones muy detallada que controla la electrónica de la máquina. Este es el dialecto natural de la máquina. Muy pocos programas se escriben actualmente en lenguaje de máquina por dos razones importantes: primero, porque el lenguaje de máquina es muy incómodo para trabajar (está formado por unos y ceros), mientras que lenguajes más avanzados son más parecidos al lenguaje humano. En segundo lugar el lenguaje máquina o ensamblador dependerá de características del ordenador en concreto, cambiando al cambiar el hardware, mientras que otros lenguajes de programación son capaces de adaptar los mismos programas a distintos procesadores.

Lenguaje de programación:

Sistema de símbolos y reglas que permite la construcción de programas con los que el ordenador puede operar así como resolver problemas de manera eficaz.

Estos contienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada / salida, cálculo, manipulación de textos, lógica / comparación y almacenamiento / recuperación.

Los lenguajes de programación se clasifican en:

- *Lenguaje Máquina:* Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje máquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 ó 1)
- *Lenguaje de Bajo Nivel (Ensamblador):* En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- *Lenguaje de Alto Nivel:* Los lenguajes de programación de alto nivel (Python, Java, Ruby...) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general en inglés), lo que facilita la escritura y comprensión del programa.

Algoritmo:

La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhwarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

La solución a cualquier problema informático involucra la ejecución de una serie de acciones en orden específico. Un procedimiento para resolver un problema en términos de: a) Las acciones a ejecutarse y b) el orden en el cual estas acciones deben ejecutarse se llama algoritmo.

Programa:

Formalmente podríamos definirlo como la expresión de un algoritmo en un lenguaje adecuado para que pueda ser ejecutado por un ordenador.

Una definición menos académica pero útil sería una secuencia de instrucciones mediante las cuales se ejecutan diversas acciones de acuerdo con los datos que se estén procesando.

Los datos y operaciones básicas.

A. identificador. Un identificador es una serie de caracteres formados por letras, dígitos y el carácter subrayado (`_`) que no inicie con dígito,

Será el nombre que damos a todo lo que manipulamos dentro de un programa (variables, constantes, funciones, etc). Por ejemplo variables, constantes, funciones, tipos definidos por el usuario etc.

B. Tipos de datos. Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como b, un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.

Hasta ahora en el curso hemos estudiado los valores de tipo entero **int()** y de tipo cadena de caracteres **str()** pero pronto aparecerán más

C. Variables.

Una variable es un identificador que puede tomar diferentes valores y donde almacenaremos la información necesaria en el programa. Puede cambiar de valor durante la ejecución del programa

Una variable es una parte de la memoria donde se puede almacenar un valor para uso de un programa.

D. Operadores

Un operador es un símbolo que indica al compilador que realice manipulaciones lógicas o matemáticas específicas.

Hasta ahora en el curso hemos aprendido los matemáticos `+` `-` `*` `/` `//` `**` y `%`.

Tiene una especial importancia el operador de asignación, `=` que empleamos para almacenar datos en variables

E Función

Definiremos función como una pequeña parte de un programa que realiza una tarea particular, a sea independientemente o, en el contexto de un programa mayor, devolviendo un resultado.

Las funciones pueden estar ya definidas en el lenguaje de programación o pueden ser definidas por el usuario.

2.- Fases para la creación de un programa.

A Definición del Problema

Conseguir una definición clara y precisa de lo que se pretende que haga el programa. Puede no ser necesario trabajar mucho en ella si se trata de un problema académico muy bien planteado, pero en general, cuando queramos nosotros mismos programar algo que planteamos nosotros mismos tenemos que tener muy claro que es lo que queremos que haga nuestro problema.

En esta fase puede ser muy útil recoger varios ejemplos de los datos que le daremos al programa y cual será el resultado esperado.

B Análisis del Problema.

Una vez que se ha comprendido lo que se desea del programa , es necesario definir:

- Los datos de entrada.
- Cual es la información que se desea producir (salida)
- Los métodos y fórmulas que se necesitan para procesar los datos.

Es importante en este punto recordar que el ordenador carece por completo de sentido común, y cosas que a nosotros nos parecen triviales (por ejemplo, una clase no puede tener un número negativo ni fraccionario de alumnos) tendrá que estar contemplado en el programa

C Diseño del Algoritmo.

Debemos definir cuales serán los pasos que debe ejecutar el ordenador para resolver nuestro problema. Posteriormente en el curso abordaremos como realizar esto en problemas complejos.

Las características de un buen algoritmo son:

Debe tener un punto particular de inicio.

Debe ser definido, no debe permitir dobles interpretaciones.

Salvo que el programa sea extremadamente sencillo es muy conveniente realizar esta tarea empleando papel y bolígrafo, puesto que hay detalles de los que no seremos conscientes hasta verlo escrito

D Codificación

En este momento escribiremos realmente nuestro programa, empleando diversas herramientas informáticas y teniendo al final nuestro código

E Prueba y Depuración

Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama *depuración*.

F Documentación

Se debe realizar un resumen de cómo funciona nuestro programa, ya sea para que otras personas puedan utilizarlo o para que cuando más adelante tengamos que usarlo o modificarlo no dependamos de nuestra memoria para recordar como funcionaba

Por una parte es una buena costumbre emplear comentarios en el código para poder guiar al lector sobre como funciona.

En proyectos grandes es muy conveniente generar un documento de texto en el que se expliquen los datos básicos, el uso de variables y en general el uso del programa

3.- Buenas prácticas en programación

Ahora que ya hemos dado nuestros primeros pasos en la programación, hemos de centrarnos en escribir programas legibles y reutilizables. Para ello es preferible usar una serie de convenciones a la hora de programar

De las variables

Los nombres de las variables deben ser descriptivos. Aunque puede parecer muy buena idea usar nombres alfabéticos consecutivos (a,b,c...) cuando haya que integrar nuestro código en programas mayores serán muy difíciles de leer, y por tanto de verificar.

Se suelen emplear dos convenciones al nombrar variables de mas de una palabra.

Uso de guiones bajos como separador: Las palabras se separan con un guion bajo como en `alumnos_ingles`

CamelCase Se pone en mayúscula la primera letra de cada palabra que no sea la primera, que se deja en minúsculas `alumnosIngles`.

En general es buena práctica añadir un comentario al definir la variable diciendo para que se empleará.

Del código

En general no se debe emplear una línea de presentación de resultados para calcular esos mismos resultados. Es una mejor estructura calcular los resultados y almacenarlos en una o varias variables que realizar el cálculo en la misma línea que se presenta

Es decir, no es adecuado

```
print( a+b)
```

Y se debe emplear

```
c= a+ b
```

```
print( c)
```

Esto nos permitirá reutilizar más fácilmente nuestro código