

Manual de Usuario

# Japson

Por Jorge Moya

Japson es un lenguaje que corre y compila en el navegador.

La sintaxis de Japson se basa en:

- Una función void main().  
La función principal del programa y es lo primero que corre.
- Otras funciones antes del main.  
Se pueden llamar otras funciones desde el main, pero tienen que ser declaradas antes del main.
- Existe un global antes de toda función en el que se pueden declarar variables.

Siempre empieza con una declaración del programa:

```
program test;
```

Después se declaran variables globales:

La estructura para declarar variables es:

```
var tipo id;
```

También se pueden declarar como:

```
var tipo id, tipo id, tipo id, ...;
```

Los tipos de ids son:

```
int, float, string, boolean
```

Para declara una función:

```
function void main() { código }
```

```
function int sum() { código }
```

```
function int sum(int x, int y) { código }
```

Cada función cuenta con tres bloques importantes.

- PARAMETROS

En caso de tener parámetros, no es necesario. Los parametros se declaran con la función.

```
function int sum(int x) { código }
```

- VARS

Variables declaradas antes de correr algún estatuto. Se tienen que declarar primero.

La estructura para declarar variables es:

```
var tipo id;
```

También se pueden declarar como:

```
var tipo id, tipo id, tipo id, ...;
```

Los tipos de ids son:

int, float, string, boolean

## ● ESTATUTOS

Hay distintos tipos de estatutos que se pueden llamar.

- Asignación  
`assign id = 5 + 10;`
- Write  
`write(id); write(5 + 10);`
- Condición  
`if(10 > 5) {  
 write(true);  
} else {  
 write(false);  
}`
- Ciclo  
`while(id) {  
 ...  
}`
- Return  
Si la función espera un retorno..  
`return id;`

Se anexa varios ejemplos de código:

Ejemplo 1:

```
program factorial;  
function int factorial(int x) {  
    if(x==0) {  
        return 1;  
    }  
    return x * factorial(x-1);  
}  
function void main() {  
    write(factorial(7));  
}
```

Ejemplo 2:

```
program test;
var int a, int b, float f;
function int uno(int c) {
    if(c > 10) {
        assign c = 10;
        return c;
    }
    return 0;
}
function void main() {
    assign a = 20;
    assign b = uno(a);
    write(b);
}
```