

## Leer/Escribir

```
Escribir U.print("..."); o U.println("...");
```

Leer datos del usuario:

```
int x = U.readInt();
```

```
double y = U.readDouble();
```

Escribir y leer dato del usuario:

```
int e = U.readInt("Ingrese su edad");
```

## Funciones

Definir función: `public static tipoRetorno nombre(parámetros) { ... }`

Tipos de retorno: int, double, boolean (true o false), void (nada)

- Ejemplo: `public static int max(int x, int y) { ... }`

Usar función: `tipo x = nombre(parámetros)`

- Ejemplo: `int x = max(a,b);`

## Funciones de Math (algunas)

`Math.abs(x)`: valor absoluto de x

`Math.pow(x,y)`:  $x^y$

`Math.sqrt(x)`: raíz cuadrada de x

`Math.sin(x)`, `Math.cos(x)`, `Math.tan(x)`

`Math.max(x,y)`: mayor entre x e y

`Math.min(x,y)`: menor entre x e y

`Math.random()`: número al azar en [0,1[

**Ojo:** num al azar entre x e y:  
`x + (int)(Math.random()*(y-x+1));`

## if/else

```
if (condición) { ... instrucciones ... }
```

```
else { ... instrucciones ... }
```

Elegir entre muchas opciones: `if / else if / else if / ... / else`

```
if (mes == 1) return 31;
```

```
else if (mes == 2) return 28;
```

```
...
```

```
else if (mes == 11) return 30;
```

```
else return 31;
```

## while y for (ciclos, repetir instrucciones)

```
while (condición) { ... instrucciones ... }
```

```
while (true) {  
  ... instrucciones ...  
  if (condición) break;  
}
```

```
for(inicialización; condición; reinicialización){  
  ... instrucciones...  
}
```



```
{  
  inicialización;  
  while(condición) {  
    instrucciones;  
    reinicialización;  
  }  
}
```

Patrones de while:

1. leer – while – procesar – leer
2. while(true) – leer – break? – procesar

## int – double

int: número entero, double: número real

Operadores: +, -, \*, /, % (a%b: resto de a/b)

División de dos números enteros es entero (3/4 es 0!)

Si uno de los operandos es real, resultado es real

## boolean

valor de verdad: true o false

Condiciones simples: <, <=, >, >=, ==, !=

Condiciones compuestas: && (and), || (or), !(not)