

Cheatsheet - Iniciación a la Programación

ACM UPM

29 de septiembre de 2017

1. Tipos

<i>tipo</i>	<i>bits</i>
int	32
float	32
double	64
char	16
Boolean	true false

Declarar una variable (*[] significa que puede estar o no*):
`tipo nombre_variable [= valor];`

2. Operadores

		Aritméticos
+	+=	<code>variable++</code>
-	-=	<code>variable-</code>
*	*=	
/	/=	
%	%=	
		Lógicos
&&		
		Relacionales
==	!= <= >=	

3. Condicionales

También conocido como “control de flujo”. Permite cambiar *qué* se ejecuta según ciertas *condiciones*.

```
if (condición) {  
    haz esto;  
}
```

```
if (condición) {  
    haz esto;  
}else{ // Si no se cumple la condición  
    haz lo otro;  
}
```

```
if (condición){  
    haz esto;  
}else if(condición2){ //Si no, si...  
    haz esto otro;  
}
```

Los condicionales se pueden anidar *ad infinitum*. Equivalencia en condicionales:

```
if (condición) {  
    if (condición2){ }  
}  
  
if (condición && condición2) { }
```

4. Bucles

```
for ([declarar var];[cond];[op tras bloque]){}  
while([condición]){}
```

Serían válidos por tanto:

```
for(;;){}  
while(true){}
```

Correspondencia entre bucles:

```
for (int i=0;i<10;i++){  
    System.out.println(i);  
}  
  
int i=0;  
while(i<10){  
    System.out.println(i);  
}
```

5. Funciones

```
(tipo|void) nombre ([argumentos]){  
int duplicar (int x) { return x*2; }  
void no_devuelve () { System.out.println("<3"); }
```

La función puede tener *modificadores* como: `public`, `private`, `static`. Esto queda fuera del temario de Prog I.

6. Anexo

13	CR	71	G	100	d
32		72	H	101	e
40	(73	I	102	f
41)	74	J	103	g
42	*	75	K	104	h
43	+	76	L	105	i
44	,	77	M	106	j
48	0	78	N	107	k
49	1	79	O	108	l
50	2	80	P	109	m
51	3	81	Q	110	n
52	4	82	R	111	o
53	5	83	S	112	p
54	6	84	T	113	q
55	7	85	U	114	r
56	8	86	V	115	s
57	9	87	W	116	t
65	A	88	X	117	u
66	B	89	Y	118	v
67	C	90	Z	119	w
68	D	97	a	120	x
69	E	98	b	121	y
70	F	99	c	122	z

Cuadro 1: Mini-tabla ASCII