# Bifurcación y Bucle

Guillermo Durán González

2022-10-04

# Bifurcación If

Siempre que la condición sea TRUE (!=0 )

```
if(3 > 2) print(' :D ')
```

```
## [1] " :D "
```

```
if(3 < 2) print(' :( ')
if(5) print(' :D ')
```

```
## [1] " :D "
```

Se pueden usar las { } para insertar varios comandos juntos:

```
if(3 > 2)
  {
  print('Es verdadero!')
  print(' :D ')
  }
```

```
## [1] "Es verdadero!"
## [1] " :D "
```

# Se ejecuta cuando la sentencia es TRUE o FALSE

```
x <- 1:10
clasif <- ifelse(x > 5, 'grande', 'chico')
clasif <- paste(x, clasif)
clasif
```

```
##  [1] "1 chico"   "2 chico"   "3 chico"   "4 chico"   "5
##  [7] "7 grande"  "8 grande"  "9 grande"  "10 grande"
```

# Manera alternativa

```
x=1
z=5
if(x>3) { y <- 10} else {y<-0}
```

# Bucle For

```
for(i in 1:10)
  {
  print(i) #Imprime consecutivamente del 1 al 10
  }
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

# Con arreglo

```r
x<-c("a","b","c","d")
for(i in 1:6)
  {
  print(x[i]) # Imprime cada uno de los elementos de x.
  }
```

```
## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"
## [1] NA
## [1] NA
```

```r
  # Los dos últimos, al no existir en x aparecen como "NA"
```

# Con arreglo

```r
x<-c("a","b","c","d")
for(letra in x)
  {
  print(letra) # Imprime cada uno de los elementos de x.
  }
```

```
## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"
```

```r
  # no es necesario cono cer el número de lementos
```

# While

```
x <- 1
## valor inicial...
while(x < 11)
  { # La condición es que x sea menor a 11
  print(x)
  x <- x + 1 ## Aumento el valor de x de a 1 por iteración
  }
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

# Cerca de Newton

```
f <- 5 # Valor inicial
n <- 0
while(f > 0.001)
{
  n <- n + 1
  f <- f / n
  print(f)
}
```

```
## [1] 5
## [1] 2.5
## [1] 0.8333333
## [1] 0.2083333
## [1] 0.04166667
## [1] 0.006944444
## [1] 0.0009920635
```

# Loops con Repeat

```
x <- 1
repeat
{
  print(x)
  x = x+1
  if (x == 6)
  {
    break
  }
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```
v <- c("Hola","mundo")
```

# Llegamos a las funciones

Función de Gottfried Leibniz para el cálculo de pi

```
leib <- function(n)
  { x <- 0:n
  imp <- 2 * x + 1
  inv <- 1 / imp
  elementos <- inv*(- 1)^x
  sum(elementos)
  }

leib(10000000)
```

```
## [1] 0.7853982
```

# Sucesión de Fibonacci

```
fibo=function(n)
  {
  Res=numeric(n)
  if(n==1)
    { Res[1]=1}
  if(n==2){ Res[1:2]=c(1,1)}
  if(n>2)
    {
    Res[1:2]=c(1,1)
    for(i in 3:n)
      {
      Res[i]=Res[i-1]+Res[i-2]
      }
    }
  Res
  }


##Ejemplo
```