Presentación

Programación en Julia: Primeros pasos

Variables, tipos y operaciones

Héctor Medel

Benjamín Pérez

Existen distintos tipos de datos en Julia

Por ejemplo: Char, Int64, Float64

```
y = 7
```

```
Int64
```

```
typeof(y)
```

```
x = "CADI"
```

```
x = "CADI"
```

```
String
```

```
typeof(x)
```

```
w = 2.5;   # Agregamos ; para que no imprima el resultado
```

```
Float64
```

```
typeof(w)
```

```
UndefVarError: z not defined
```

1. top-level scope @ Local: 1

```
y+z
```

```
MethodError: no method matching +(::Int64, ::String)
Closest candidates are:
+(::Any, ::Any, !Matched::Any, !Matched::Any...) at operators.jl:591
+(::T, !Matched::T) where T<:Union{Int128, Int16, Int32, Int64, Int8, UInt128, UInt16, UInt32, UInt64, UInt8} at int.jl:87
+(::Union{Int16, Int32, Int64, Int8}, !Matched::BigInt) at gmp.jl:537
...
```

1. top-level scope @ Local: 1 [inlined]

```
y+x
```

```
9.5
```

```
y+w
```

Algunas ideas para nombrar variables

- Minúsculas con múltiples palabras separadas por un guion bajo (_).
- Nombres cortos.
- Podemos usar símbolos Unicode. Por ejemplo, **\alpha[TAB]** despliega α.

```
current_time = 0.7
```

```
current_time = 0.7
```

```
α = 2
```

```
α = 2
```

Los comentarios se agregan usando el símbolo **#**

```
β = 3.1
β = 3.1 # Este es un comentario
```

Para desplegar valores usamos la funcion **print()**

```
print(β)
```

```
3.1
```

Incluso podemos agregar texto y el valor de una variable usando el mismo comando

```
print("El valor de la variable β es $(β)")
```

```
El valor de la variable β es 3.1
```