

Programming basics

2024-04-16

TUDÚ

- comments
- packages
- functions vs methods (multiple dispatch)
- meaning of dynamic typing
- custom types
- broadcasting of functions
- array comprehensions
- explicit type declarations for custom types and function arguments
- conditionals
- loops

Variables and assignments

- In programming, a **variable** is a “storage box” that stores data for later use
- The data is **assigned** to the variable using the = operator
- Here, we assign the number 5 to a variable named `my_number`:

```
my_number = 5
```

- We can now do things such as:

```
my_number + my_number
```

10

Fundamental types

- Variables can store different **types** of data:
 - Integers: 1, 2, -100, ...
 - Floating-point numbers (“floats”): 3.14, pi, 1.0, ...
 - Booleans: true, false
 - Strings: "John", "Mary"
 - Arrays: [1, 2, 3, 4], [1 2 3 4]
 - And some others... we’ll meet them later

Arithmetic operations

- Arithmetic operations are mostly self-explanatory. For example:

```
number1 = 15
number2 = 20
number3 = 10*(number1 + number2) - number1/number2
number3
```

349.25

String concatenation

- Julia overloads the * operator for strings too:

```
string1 = "This "
string2 = "is a"
string3 = " sentence"
string1 * string2 * string3 * "!"
```

"This is a sentence!"

Arrays

- An array is a (possibly multidimensional) collection of objects
 - A one-dimensional array is a vector, a two-dimensional array is a matrix, and so on
- Usually we work with arrays of numbers. They are easy to create:

```
my_array = [10, 20, 30, 40]
```

```
4-element Vector{Int64}:
```

```
10  
20  
30  
40
```

Accessing array contents

- The elements of an array can be accessed one-by-one by referencing their location or **index** in the array:

```
my_array = [10, 20, 30, 40]  
my_array[1]
```

```
10
```

or

```
my_array[2]
```

```
20
```

- The special keyword **end** fetches the last element:

```
my_array[end]
```

```
40
```

- Arrays can also be subsetted:

```
my_array[2:3]
```

```
2-element Vector{Int64}:
```

```
20  
30
```