Module 2: Expressions

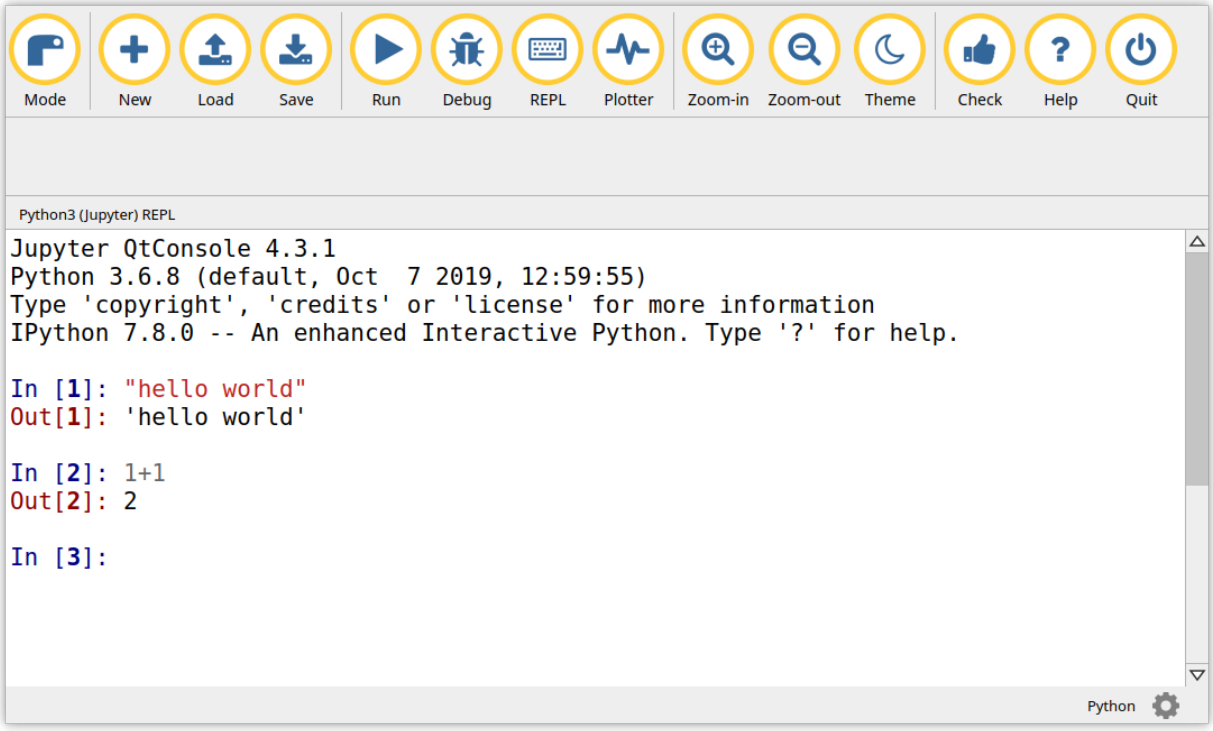
An expression is a combination of program elements that the computer will evaluate to a value.

Here are some valid Python expressions:

- 2 + pi
  - *that's a constant number (2), the addition operator (+), and a variable (pi)*
  - this evaluates to 5.14159265...
- "cup" + "cakes"
  - this evaluates to the word "cupcakes"
- 42
  - this evaluates to (duh!) 42

REPL

REPL stands for: read - evaluate - print - loop



Math Operators

Operators are used to build expressions, like “1 + 1”:

Operator	Character Name	Operation	Order
`( … )`	parenthesis	order of operation	1 (highest)
`**`	–	exponentiation	2
`+`	plus	addition	3
`-`	minus	subtraction	
`/`	slash	division	4
`*`	asterisk	multiplication	

Math Operator Exercises

Using the REPL, determine what the following expressions evaluate to:

Expression	Evaluation Result
`3`	
`3 + 3`	
`3 - 3`	
`3 * 3`	
`3 / 3`	
`1 / 4`	
`3 ** 2`	
`2 + 3 * 3`	
`(2 + 3) * 3`	
`1 + 10 / 2`	
`(1 + 10) / 2`	

⁉ **QUESTION:** Which expressions were *readable* (easy to figure out) and which were not?

Value Types

Values also have a type. Some common types are:

Type	Description	Example
str	String value.	“Hello world”
int	Integer number (or whole number) value.	3
float	Floating point number (or decimal number) value.	0.25

Python has functions for getting or changing a value type. Test the following code fragments in the REPL:

Fragments	Evaluation Result
`a = "hello world"` `type(a)`	
`b = 1+1` `type(b)`	
`c = 1/2` `type(c)`	

`d = "99"`
`type(d)`
`e = int(d)`
`type(e)`	

`f = "0.25"`
`g = float(f)`
`type(g)`	

`int("hello")`	
`int(4.9)`	
`int(-4.9)`	
`2 + "2"`	
`2 + int("2")`	

One last new function to try in the REPL:

Fragments	Evaluation Result
`x = 1.8933` `int(x)`	
`round(x)`	
`round(x, 1)`	
`round(x, 2)`	

Programming Challenge (dogyears.py)

Write a program that does the following when run:

```
How old are you? 17
Your age in dog years is: 2.4
>>>
```

Important notes:

- To start, close the REPL if you haven't already. Open a NEW tab and name it “dogyears.py”.
- Let’s use the old rule of thumb that 7 people years = 1 dog year. That's not really accurate, but go with it!