Lecture 1: Expressions

CS 51P

August 31, 2022

Terminology

- Value
- Type
- Operator

Types

A **type** is a set of values and plan for representing/interpreting those values in binary

int

- Values: 0, 1, -10, 34022, …
- Operations: +, -, /, *
  - ** (exponent),
  - % (remainder)
  - // (truncated division)

string

- Values: "Hi!", "", "2.0",…
- Operations: + (concatenation)
  - * (duplication),

All values have types
Common types: int, float, str, bool

Terminology

- Value
- Type
- Operator
- Expression

Expressions

ex·pres·sion

/ikˈspreSHən/

noun

noun: **expression**; plural noun: **expressions**

1. the look on someone's face that conveys a particular emotion.
   "a sad expression"
   synonyms: look, appearance, air, manner, countenance, mien
   "an expression of harassed fatigue"

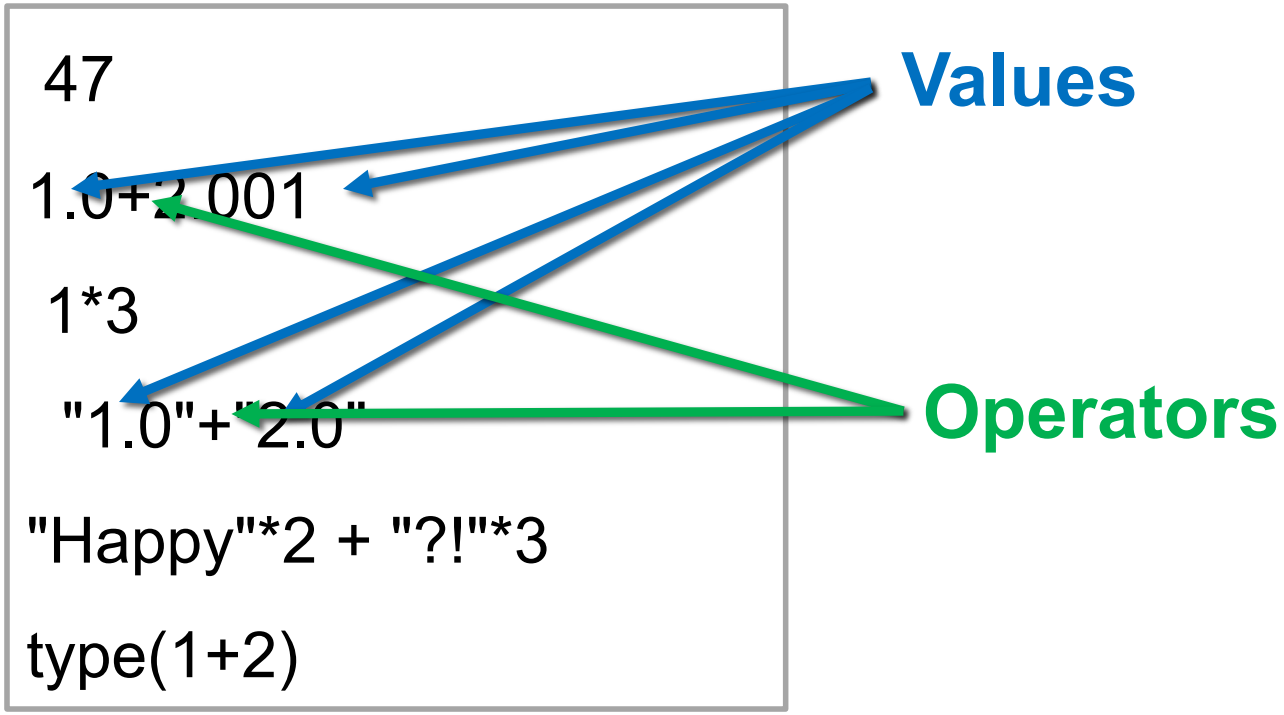
2. a word or phrase, especially an idiomatic one, used to convey an idea.
   "nowhere is the expression “garbage in, garbage out” any truer"
   synonyms: idiom, phrase, idiomatic expression; More

   - MATHEMATICS

     a collection of symbols that jointly express a quantity.
     "the expression for the circumference of a circle is 2πr"

Expressions

- Expressions represent a value
- Python evaluates expressions (similar to a calculator)



High-level languages

Figure 1.1: An interpreter processes the program a little at a time, alternately reading lines and performing computations.

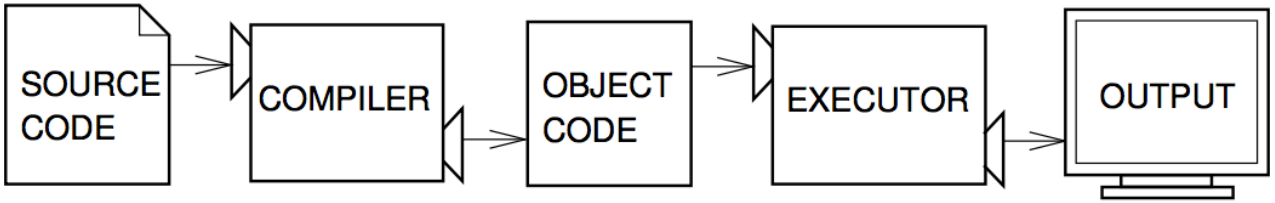


Figure 1.2: A compiler translates source code into object code, which is run by a hardware executor.

Machine Language

```
lcfi2:
        movl    %edi, -4(%rbp)
        cmpl    $0, -4(%rbp)
        jle     LBB0_2
## BB#1:
        leaq    L_.str(%rip), %rdi
        movb    $0, %al
        callq   _printf
LBB0_2:
        xorl    %eax, %eax
        retq
L_.str:
        .asciz "x is a positive number"
```

Exercise 1: Expressions

```
1/2

13

4 + 3 * 2

( "A"*2 + "?"*3 ) * 2

14 % 5

5 ** 2

Hi!

1*2 + "2"*2
```

Errors

- Two types of errors:
  - SyntaxError: invalid syntax
  - TypeError: unsupported operand type(s) for +: 'int' and 'str'

```
>>> Hi!
  File "<input>", line 1
    Hi!
      ^
SyntaxError: invalid syntax
>>> 1*2 + "2"*2
Traceback (most recent call last):
  File "<input>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Casting

- You can change types by casting
  - str( ), int( ), float( )

```
1*2 + 2"*2
```

```
1*2 + int("2")*2

str(1)*2 + "2"*2
```

- You can always check the type of a value (or expression)n using the command type()

```
type(1*2)

type("2"*2)
```

Exercise 2: Expressions and Errors

```
3 * "5"
str(3) * int("5")
1 / 2 * "Hello"
"1.0" + 2.0
int("2") * "2"
1 ** 2.5
str("2") * 4
```

Assigning variables

- Can assign a value to a variable
- Right hand side can be any expression (anything that is, or that evaluates to, a value)

```
x = 13

a_string = 1*str(2) + "2"*2

x_type = type(1+2.001)
```

Variables and Expressions

- a variable evaluates to the value stored in that variable
- variables can be used in expressions

```
my_num = 13

new_num = my_num + 34
```

Example: Writing a Program

If you run a 10 kilometer race in 43 minutes 30 seconds, what is your average time per mile? (Hint: there are 1.61 kilometers in a mile).

Exercise 3: Writing a Program

If you run a 10 kilometer race in 43 minutes 30 seconds, what is your average speed in miles per hour? (Hint: there are 1.61 kilometers in a mile).