GENERAL ASSEMBLY

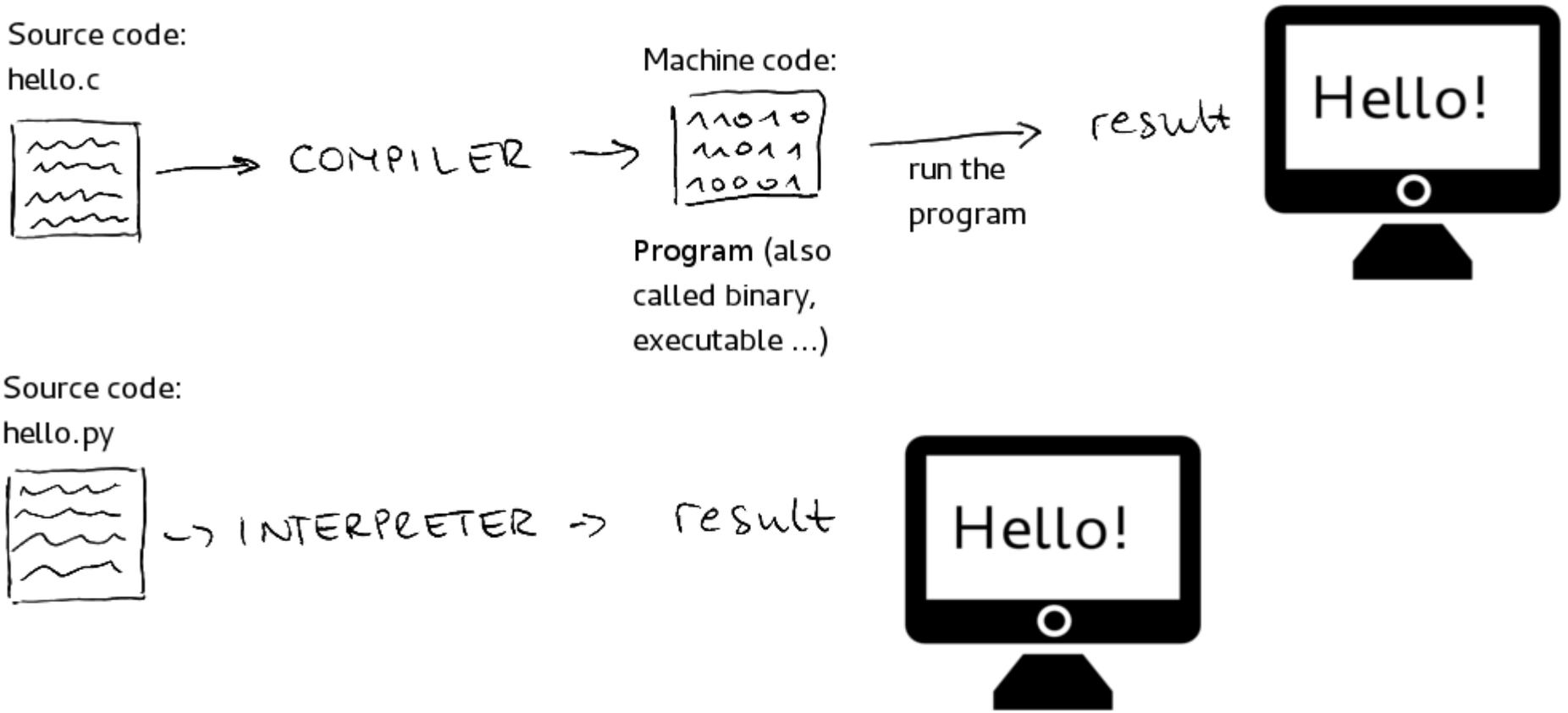
INTRO TO PYTHON FUNDAMENTALS

Ramkumar Hariharan, PhD
Scientist,
Institute for Systems Biology,
Seattle

Why did van Rossum call it Python ?

1) He liked Pythons and thought they were just awesome
2) He was a fan of “Monty Python’s Flying Circus”
3) He was well, a bit batty, that’s all !
4) I don’t know, and after all, who really cares ?

What kind of a language is Python



Source: ikajic.github.io

Why Python ?

“Hello world” in Java

```
public class HelloWorld {
   public static void main(String[ ]
args) {
      System.out.println("Hello,
World");
   }
}
```

“Hello world” in Python

```
print (“Hello World”)
```

Python is an expressive language: each statement says a lot !

Ratio of High-Level-Language Statements to Equivalent C Code

Language	**Level Relative to C**
C	1
C++	2.5
Fortran 95	2
Java	2.5
Perl	6
Python	6
Smalltalk	6
Microsoft Visual Basic	4.5

Source: Adapted from *Estimating Software Costs* (Jones 1998), *Software Cost Estimation with Cocomo*

Higher ratio = higher productivity !

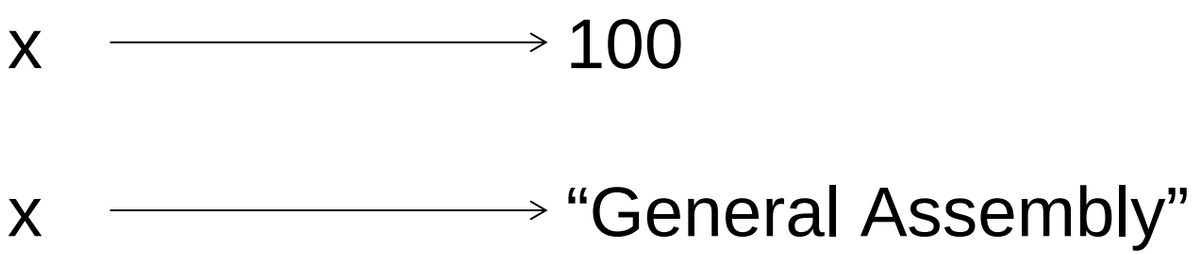
Five fun facts about Python

1. A dynamic, interpreted language
2. No type declarations
3. Error checks at runtime
4. Python source code, or “scripts” are modules…end in .py
5. Indentation is used to demarcate blocks of code

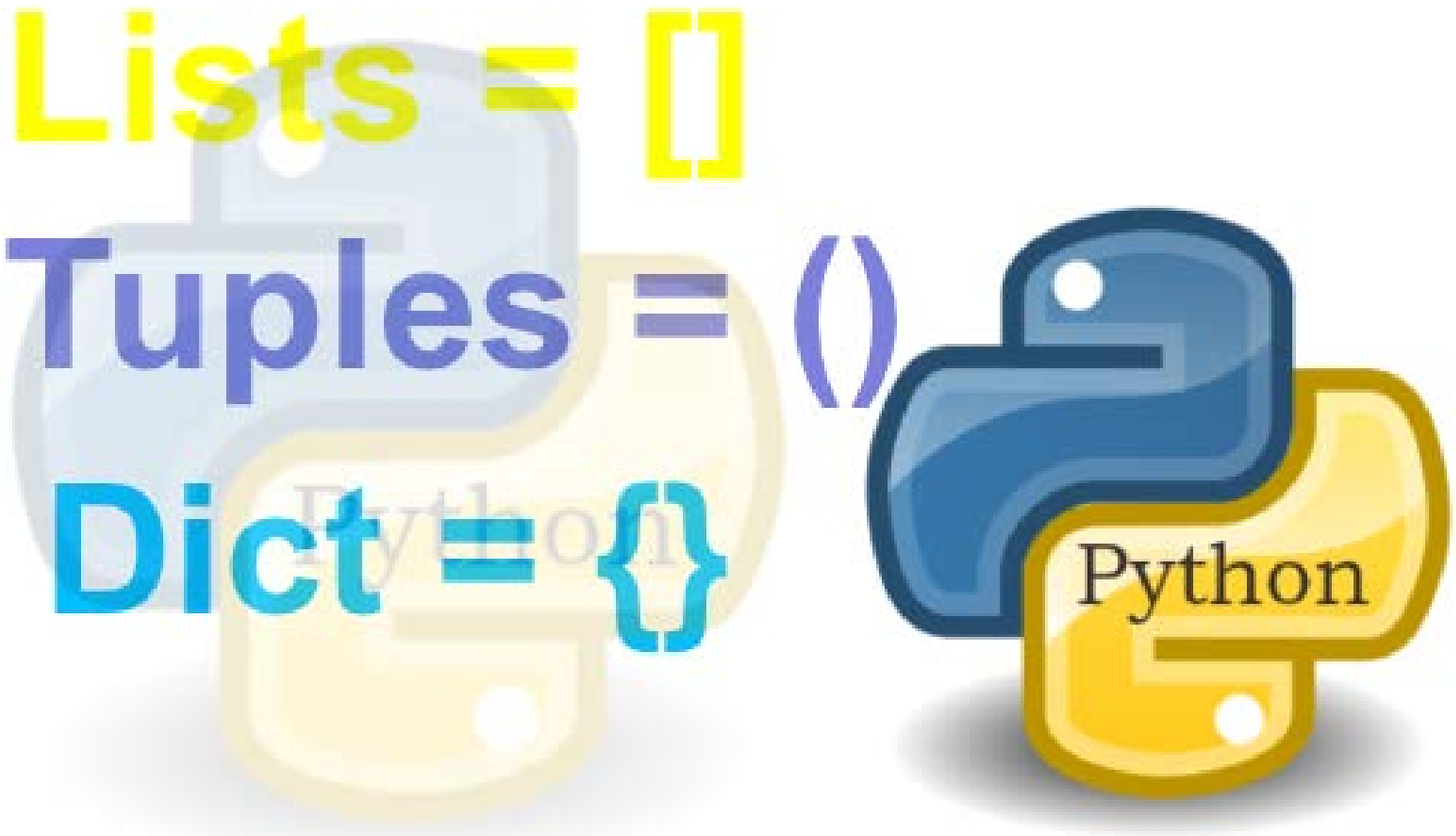
Our punch line for this class

You can learn one fun ton of Python in 4 hours !

How to think of variables in Python

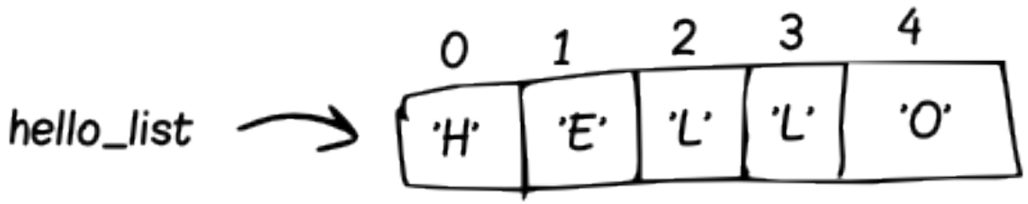


Common data structures in Python



Source: teknosains.com

Python lists



- Python lists implemented via indirection
- Lists enclosed with [ ]
- Lists are mutable

Source: cvcconline.org

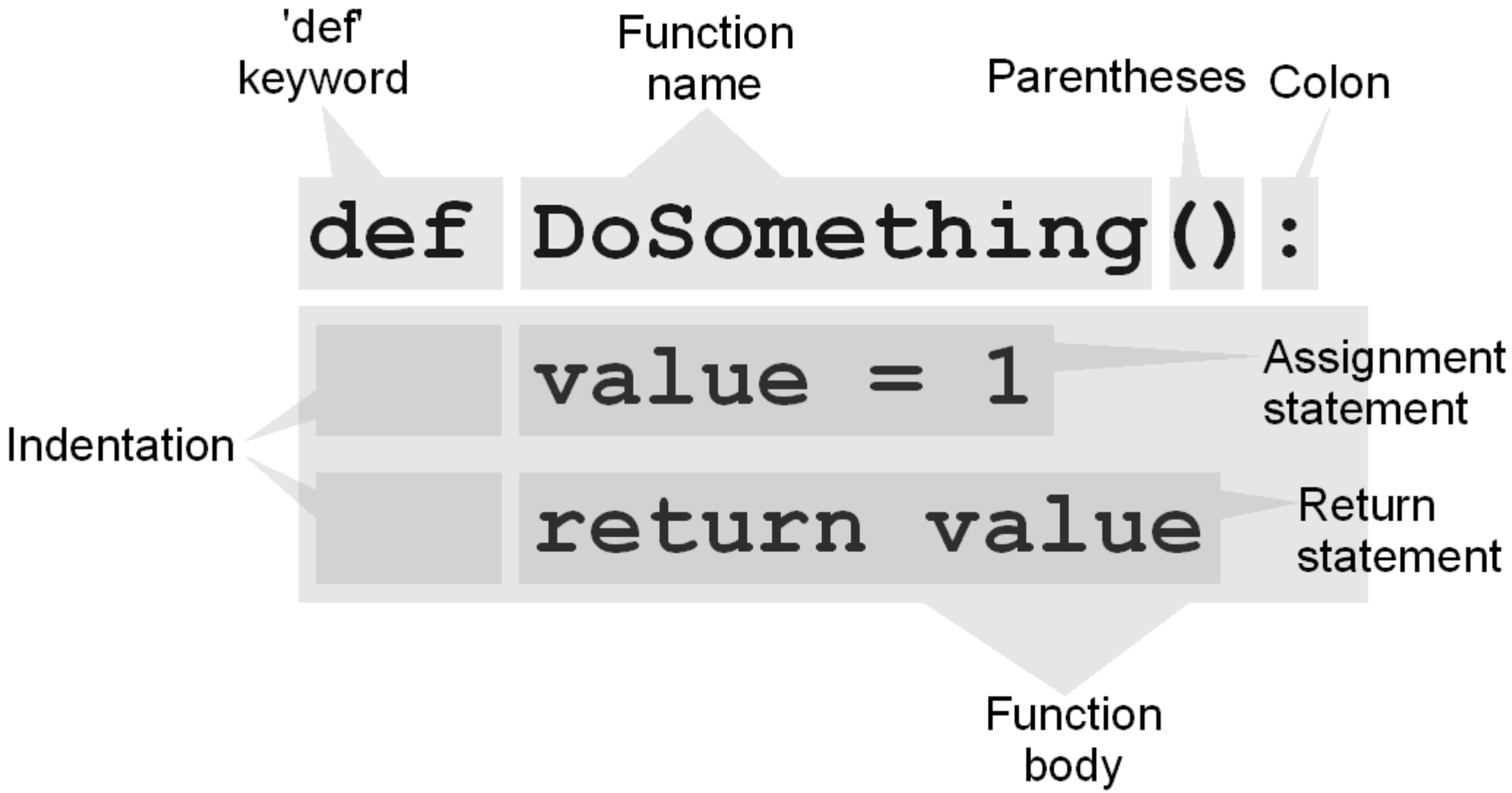
Python tuples

- Enclosed with ()
- Tuples are immutable
- e.g. mytuple = (1,2,3,4)

Python dictionaries

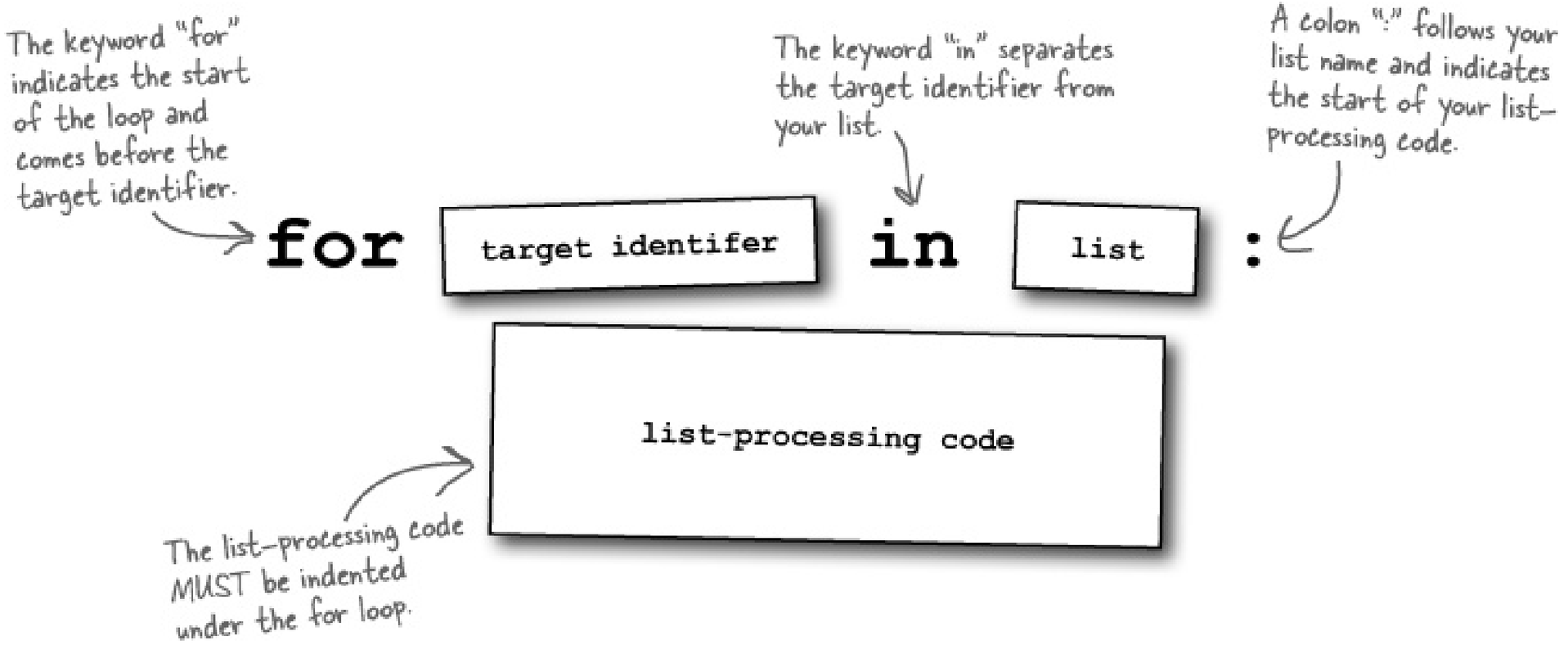
- Comprise key value pairs
- Implicitly implement hash tables
- Facilitates fast look up
- Enclosed in { }

Python functions



Source: weebly.com

For loops in Python



Source: safaribooksonline.com

Python re module: regular expressions made easy

Summary of Regular Expression Patterns

Atoms

Plain symbol:	`...`
Escape:	`\`
Grouping operators:	`( )`
Backreference:	`\#, \##`
Character class:	`[ ]`
Digit character class:	`\d`
Non-digit character class:	`\D`
Alphanumeric char class:	`\w`
Non-alphanum char class:	`\W`
Whitespace char class:	`\s`
Non-whitespace char class:	`\S`
Wildcard character:	`.`
Beginning of line:	`^`
Beginning of string:	`\A`
End of line:	`$`
End of string:	`\Z`
Word boundary:	`\b`
Non-word boundary:	`\B`
Alternation operator:	`|`

Constants

`re.IGNORECASE`	`re.I`
`re.LOCALE`	`re.L`
`re.MULTILINE`	`re.M`
`re.DOTALL`	`re.S`
`re.UNICODE`	`re.U`
`re.VERBOSE`	`re.X`

Quantifiers

Universal quantifier:	`*`
Non-greedy universal quantifier:	`*?`
Existential quantifier:	`+`
Non-greedy existential quantifier:	`+?`
Potentiality quantifier:	`?`
Non-greedy potentiality quantifier:	`??`
Exact numeric quantifier:	`{num}`
Lower-bound quantifier:	`{min,}`
Bounded numeric quantifier:	`{min,max}`
Non-greedy bounded quantifier:	`{min,max}?`

Group-Like Patterns

Pattern modifiers:	`(?Limsux)`
Comments:	`(?#...)`
Non-backreferenced atom:	`(?:...)`
Positive Lookahead assertion:	`(?=...)`
Negative Lookahead assertion:	`(?!...)`
Positive Lookbehind assertion:	`(?<=...)`
Negative Lookbehind assertion:	`(?<!...)`
Named group identifier:	`(?P<name>)`
Named group backreference:	`(?P=name)`

Source: etutorials.org

Five great Pythony resources

1. Help()
2. Dir()
3. Google
4. Python docs
5. Stackoverflow