DS2000 – Programming with Data

02. Intro to Python

John Rachlin, PhD
Northeastern University

Statements

A **statement** is a singular task to be performed.

An **assignment statement** assigns a value to a variable.

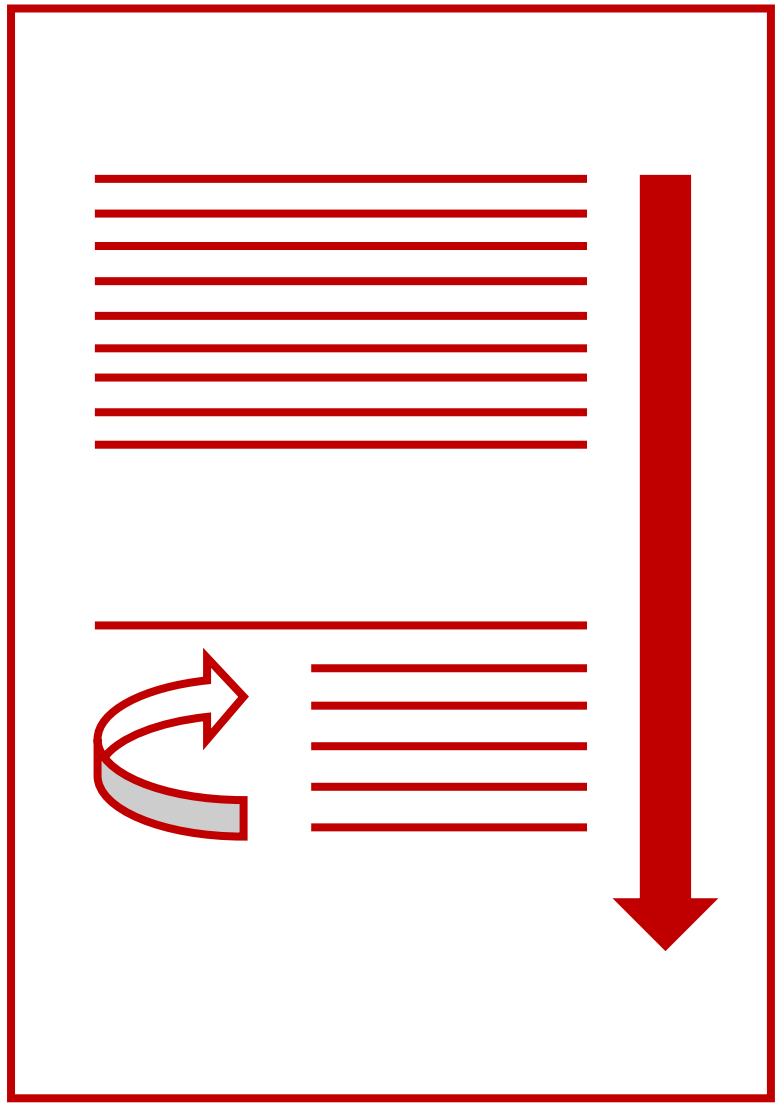
```
x = 5
msg = “hello”
total = 2 + x
pi = 3.14159
etc.
```

Programs

A **program** is a sequential series of statements that are executed
one at a time from the top of the file to the bottom.

Programs may contain **loops** that cause certain statements to execute repeatedly.

Other types of statements may impact program flow. For example, **conditional statements** may cause certain statements to be executed only under specific conditions.



Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`
Binary Types:	`bytes`, `bytearray`, `memoryview`

https://www.w3schools.com/python/python_datatypes.asp

Arithmetic operators

Python operation	Arithmetic operator	Algebraic expression	Python expression
Addition	+	$f + 7$	`f + 7`
Subtraction	-	$p - c$	`p - c`
Multiplication	*	$b \cdot m$	`b * m`
Exponentiation	**	x^y	`x ** y`
True division	/	x/y or $\frac{x}{y}$ or $x \div y$	`x / y`
Floor division	//	$\lfloor x/y \rfloor$ or $\left\lfloor \frac{x}{y} \right\rfloor$ or $\lfloor x \div y \rfloor$	`x // y`
Remainder (modulo)	%	$r \bmod s$	`r % s`

Operator precedence

Operators	Grouping	Type
`( )`	left to right	parentheses
`**`	right to left	exponentiation
`*` `/` `//` `%`	left to right	multiplication, true division, floor division, remainder
`+` `-`	left to right	addition, subtraction
`>` `<=` `<` `>=`	left to right	less than, less than or equal, greater than, greater than or equal
`==` `!=`	left to right	equal, not equal

```
2 ** 3 / 4 - 1
8 / 4 - 1
2.0 - 1
1.0
```

Built-in functions

pow(x, y): Compute x to the power of y. E.g., pow(5,2) is 25.
type(x): What is the data type of x?
print(x): Print x to the screen – multiple parameters possible
input("<prompt>"): Get user input

```
In [28]: name = input("What is your name? ")
What is your name? John

In [29]: print("Hello",name)
Hello John

In [30]:
```

Built-in functions

pow(x, y): Compute x to the power of y. E.g., pow(5,2) is 25.

pow is the name of the function.
x and **y** are function *parameters* (also called *arguments*)

Many functions, like **pow**, return a result. **pow** returns either an integer (int) or a floating-point (float) number.

Some functions, like **print**, don't return a result. But they may have side-effects, such as displaying text to the screen.

Style

Style guidelines help make your program more readable and thus easier to understand.

Your future colleagues will thank you!

Spaces: total = sum + 4 (**not** total=sum+4)
Naming: tax_credit (**not** Tax_Credit **or** taxCredit **or** tc57)

Strings

A **string** is a sequence of characters

Single or double-quoted (be consistent):

```
print(“hello world”)
print(‘hello world’)
```

Escape sequence	Description
\n	Insert a newline character in a string. When the string is displayed, for each newline, move the screen cursor to the beginning of the next line.
\t	Insert a horizontal tab. When the string is displayed, for each tab, move the screen cursor to the next tab stop.
\\	Insert a backslash character in a string.
\"	Insert a double quote character in a string.
\'	Insert a single quote character in a string.

Multiline strings

```
In [51]: haiku = """
    ...: An old silent pond
    ...: A frog jumps into the pond --
    ...: Splash! Silence again.
    ...: """

In [52]: print(haiku)

An old silent pond
A frog jumps into the pond --
Splash! Silence again.
```

Source: https://www.readpoetry.com/10-vivid-haikus-to-leave-you-breathless/

Comparison operators

Algebraic operator	Python operator	Sample condition	Meaning
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

Descriptive Statistics

Measures of *dispersion* (variability):

- Range / Interquartile range
- Variance
- Standard Deviation

Measures of *central tendency:*

- Mean
- Median
- Mode