

PYTHON CHEAT SHEET

DATA TYPES

Students need to be aware of 4 data types:

- String → "this is a string" ; you can use either single quotes ' ' or double quotes " "
- Integer → 12 ; an integer is a whole number, such as 12, 0 or -4
- Float → 3.14 ; a float is any number with decimals
- Boolean → True ; note True and False require a capital T or capital F

Colors in the Codesters text editor are related to type:

- "strings are green"
- numbers like 12 or 3.14 are blue
- Booleans like True and False are red
- variable names are orange
- everything else is white including functions, classes, punctuation, and reserved words

PYTHON CHEAT SHEET

MATH AND RANDOM

$6 + 3 \rightarrow 9$ + addition

$6 - 3 \rightarrow 3$ - subtraction

$6 * 3 \rightarrow 18$ * multiplication

$6 / 3 \rightarrow 2$ / division

$7 \% 3 \rightarrow 1$ % remainder (modulo)

$6 ** 3 \rightarrow 196$ ** exponent

`random.randint(1, 10)` → creates a **random integer** between 1 and 10

`random.rand()` → creates a **random float** (decimal) between 0 and 1

`"Hello" + " world!"` → **Hello world!** +combine two strings

`"I am " + str(13)` → **I am 13** `str()` when combining a string and an integer with +

PYTHON CHEAT SHEET

VARIABLES

Variables let you store a piece of data to use later. This data can also change over the course of a program, so it's useful for keeping track of things like scores. To create a variable, just type the name of the variable followed by an equals sign and a value. For example:

- `fav_class = "coding"`
- `score = 14`

3 rules for naming variables:

- 1) Do not put variable names in quotes
- 2) Variable names cannot contain spaces, use underscores `_` instead
- 3) Variable names cannot begin with numbers

For Example:

- `"result" = 12` ; must be → `result = 12`
- `my name = "Codesters"` ; must be → `my_name = "Codesters"`
- `3rd_example = True` ; must be → `third_example = True`

PYTHON CHEAT SHEET

BLOCKS (If statements, Loops, Functions, and Events)

If Statements let you execute or skip a group of commands depending on whether the condition you provide is true or false.

- For example – this tests whether you guess the correct answer “blue”.

```
if guess == "blue":  
    sprite.say("Correct!")  
else:  
    sprite.say("Try again.")
```

- You can use the following comparisons and logical operations:

== equal != not equal

> greater than >= greater than or equal to

< less than <= less than or equal to

and or

Loops let you execute a group of commands multiple times.

- For example – this tells the sprite to count to 5.

```
for counter in range(5):  
    sprite.say(counter)
```

- For example – this tells the sprite to say each letter in “Codesters”.

```
word = "Codesters"  
for letter in word:  
    sprite.say(letter)
```

Functions let you create a group of commands that you can run later.

- For example, this function squares a number, says the square, then returns the result.

```
def square_function(number):  
    squared = number ** 2  
    sprite.say(squared)  
    return(squared)  
  
result = square_function(9)
```

Events let you create a group of commands that will run when certain things happen on the stage, or the user presses a particular button.

Whitespace and formatting blocks

- Python is **very** particular about whitespace. All code inside of these blocks must be indented four spaces.
- Codesters will highlight the whitespace in these blocks. If your highlighting is missing, check your whitespace.
- The first line of all these blocks always ends in a colon.