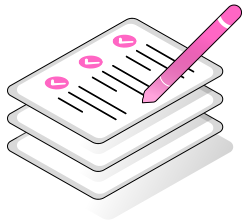
Cheat Sheet: Python

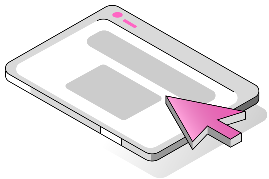


1

Types: Basic

```
int: 123
float: 1.23
bool: True / False
str: “Hello World”
str line break: “Hello\nWorld”
```

```
str multiline:
“““
Hello
World
”””
```



2

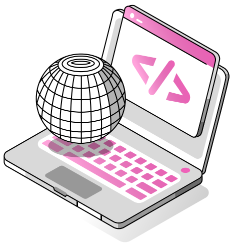
Types: Container

Ordered sequences, index access, duplicate values

```
list: [1, “2”, 1] / [”a”, “b”, 9]
tuple: (3, 2, 7) / (“d”, 800, “d”)
```

Unordered, unique keys

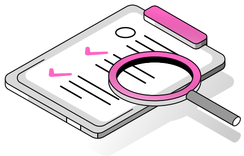
```
dict: {”key1”: “value”, “key2”: “value”, “a”: 1}
set: {”a”, 2, “hello”}
```



3

Variables

```
x = 2 + 9
a = b = c = 0 → same value assignment
d, e, f = 44, 3, 8 → multiple assignment
cheat_sheet = “python”
state = True
```

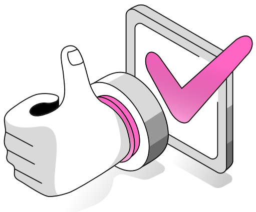


4

Indexing & Slicing

Lists, tuples, strings...

```
nums = [4, 9, 2, 5] / word = “word”
→ positive indices: 0, 1, 2, 3
→ negative indices: -4, -3, -2, -1
print(nums[0], word[-2]) → 4 r
print(nums[0:2], word[:-2]) → [4, 9] wo
```



5

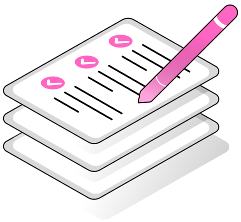
List Operations

```
nums.append(val) → add element to end of list
nums.insert(idx, val) → insert element at index position
nums.remove(val) → remove first instance of val
nums.sort() → sort list in place
nums.reverse() → invert list in place
```

lucasfortune.github.io

Cheat Sheet: Python

6



Boolean Logic

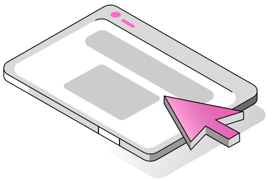
Comparators

```
<  → less than
>  → greater than
<= → less than or equal to
>= → greater than or equal to
== → equal to
!= → not equal to
```

Operators

```
a and b
a or b
not a
```

7



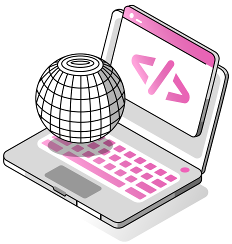
Conditions

```
if a > b:
    print(a)
elif a == c:
    print(c)
else:
    print(b)
```

```
if a:   → if a is True or not empty
    a_exists = True

if not b:   → if b is False or empty
    print(“False or empty”)
```

8



Loops

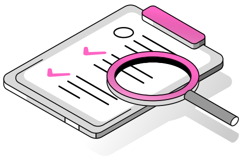
```
data = [1, 2]
for num in data:
    print(num)

→ 1
→ 2
```

```
i = 0
while i < 2:
    print(i)
    i += 1

→ 0
→ 1
```

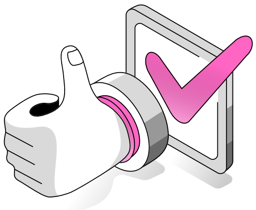
9



Dictionary Operations

```
d[“key”] → access value of “key”
d[“key”] = “value” → assign “value” to “key”
d.update({“key2”:“value2”}) → update/add key-value pair
d.keys() → return keys
d.values() → return values
d.items() → return key-value pairs
```

10

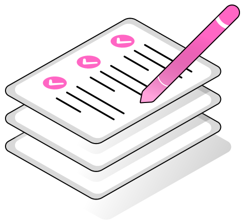


Set Operations

```
A | B    A.union(B)    →  {1, 2, 3, 4, 5}
A & B    A.intersection(B)  → {3}
A - B    A.difference(B)   → {1, 2}
     ◊ Not the same as:
     B - A    B.difference(A) →  {4, 5}
A ^ B    A.symmetric_difference(B)   → {1, 2, 4, 5}
```

Cheat Sheet: Python

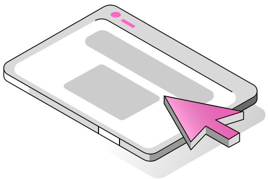
11



Functions

```
def function_name(parameter1, parameter2):
    """documentation"""
    print("function")
    c = parameter1 + parameter2
    return c
```

12

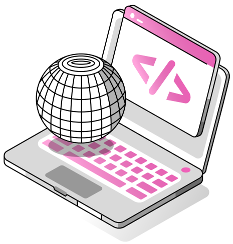


Pandas: Reading & Creating Files

```
df = pd.read_csv("data.csv")
df.to_csv("new_data.csv")

df = pd.read_excel("data.xlsx")
df.to_excel("new_data.xlsx")
```

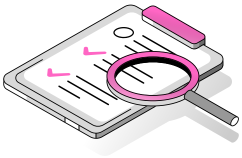
13



Pandas: Inspecting Data

```
df.head() → first rows
df.tail() → last rows
len(df) → number of rows
df.shape → dimensions
df.dtypes → variable types
```

14



Pandas: Inspecting Data

```
df.isnull() → null values
df.isna() → missing values
df.duplicated() → duplicate rows
df.info() → column names, number of rows, missing
             data, variable types
df.describe() → basic statistics
```

15

Pandas: Managing Outliers

```
df.dropna() → delete rows with missing values
df.drop_duplicates() → delete duplicate rows
df.fillna(df.mean()) → assign mean to missing values
```

lucasfortune.github.io