
| Python Quick Reference Cheat Sheet |

==========

| Basics |

==========

| Variables and Types |

Variable: A location in memory to store values.

Variable Type: Defines the type of data that the variable can hold.

Examples:

```
name = "John"       # String variable
age = 25            # Integer variable
price = 19.99       # Float variable
is_active = True    # Boolean variable
```

| Operators |

Operator | Description | Example

+ | Addition | 3 + 5 # Result: 8

- | Subtraction | 7 - 2 # Result: 5

* | Multiplication | 4 * 3 # Result: 12

/ | Division | 10 / 2 # Result: 5.0

// | Floor Division | 10 // 3 # Result: 3

% | Modulus | 10 % 3 # Result: 1

** | Exponentiation | 2 ** 3 # Result: 8

| Control Structures |

| Control Structure: Conditionals (if-else) |

```
if condition:
    # Code block if condition is true
else:
    # Code block if condition is false
```

Example:

```
age = 18
if age >= 18:
    print("You are an adult")
else:
    print("You are a minor")
```

| Control Structure: Loops (for, while) |

```
for item in sequence:
    # Code block that repeats for each item in the sequence
```

Example:

```
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num)
```

```
while condition:
    # Code block that repeats while the condition is true
```

Example:

```
counter = 0
while counter < 5:
    print(counter)
    counter += 1
```

==========

| Functions and Modules |

==========

| Functions |

Function: A reusable block of code that performs a specific task.

Syntax:

```
def function_name(parameters):
    # Function code
    return result
```

Example:

```
def add(a, b):
    return a + b

result = add(2, 3)   # Result: 5
```

| Modules |

Module: A file that contains Python definitions and statements.

Import:

```
import module
```

Usage:

```
module.function()
```

Example:

```
import math

result = math.sqrt(16)   # Result: 4.0
```

==========

| Exception Handling |

==========

```
try:
   # Code that may raise an exception
except ExceptionType as exception_name:
   # Code block to handle the exception
```

Example:

```
try:
    result = 10 / 0
except ZeroDivisionError as error:
    print("Error:", error)
```

==========