

### Introduction

#### Variable & Operator & Casting & Input

String	<code>name = " Cho pit o"</code>
Int	<code>number = 13</code>
Operator	<code>+ - * / // %</code>
Input	<code>example = input( " Who ?:")</code>
Casting	<code>number = int(in put ("is 1?:"))</code>

#### Print & Type

Print	<code>print( " nam e:", name)</code>
Printf	<code>print( f"na me: {na me} ")</code>
Type	<code>type(name)</code>

#### Conditional

IF	<code>(if elif else)</code>
MATCH	<code>match option: case 1: case _:</code>

#### Loop

For	<code>for i in range(10)   for x in y</code>
While	<code>while n &lt;= 13:</code>

#### List

List	<code>exampl e_list = ["hi ", " bye ", 13]</code>
Len	<code>len(ex amp le_ list)</code>
Position	<code>exampl e_l ist[0]</code>
Range	<code>exampl e_l ist [0:1]   exampl e[- 1: -2]</code>
In	<code>" hi" in exampl e_list</code>
Index	<code>exampl e_l ist.in dex ("by e")</code>

#### Dictionary

Dictionary	<code>numbers = {"on e": 1, " thi rte en": 13}</code>
Position	<code>number s["t hir tee n"]</code>
Key	<code>number s.k eys()</code>
Value	<code>number s.v alues()</code>
For in Dictionary	<code>for key, value in list.i tems():</code>
Add with key	<code>number s["t wo"] = 2</code>
Remove with key	<code>number s.p op( " two ")</code>
Update with key	<code>number s["t wo"] = 5</code>

### Introduction

#### Function

Function	<code>def name(a, b):</code>
----------	------------------------------

#### Compression List

List	<code>new_list = [exp for x in y if con]</code>
------	---

### OOP

#### OOP

Init	<code>def __init __ ( self, x: str = " y"):</code>
Str	<code>def __str__ (s elf): return f"y: {se lf .x }"</code>
@class-method	You can call it without a instance

#### Inheritance

Inheritance	<code>class Car(Ve hicle):</code>
Super	<code>super( )._ _in it_ _ (name)</code>

#### Override

Override	<code>from typing _ex ten sions import overri de</code>
----------	---

Protected & Private	<code>(Pro)_ (Pri)_ _</code>
---------------------	------------------------------

### Map Filter Reduce

High level orders	<code>def saluda (no mbre, funcion):</code>
-------------------	---

lambda	<code>even = lambda n: n % 2 == 0</code>
--------	--

map()	<code>list(m ap( lambda n: n % 2 == 0, lis t)</code>
-------	--

filter()	<code>list(f ilt er( lambda n: n &lt; 0, list ))</code>
----------	---

reduce()	<code>from functools import reduce</code>
----------	---

reduce() example	<code>`reduce(lambda x, y: x + y, list(filter(lambda n: n &lt; 0, list)))`</code>
------------------	---



By AdrianPerogil

Not published yet.  
Last updated 9th October, 2024.  
Page 1 of 1.

[cheatography.com/adrianperogil/](https://cheatography.com/adrianperogil/)

Sponsored by [ApolloPad.com](https://apollopad.com)  
Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>