

Python Cheat Sheet – Part 2

1. Built-in Functions

Category	Function	Applies To	Example
Type Conversion	<code>int()</code>	str, float, bool	<code>int('5') → 5</code>
	<code>float()</code>	str, int, bool	<code>float('3.14') → 3.14</code>
	<code>str()</code>	any type	<code>str(5) → '5'</code>
	<code>bool()</code>	any type	<code>bool(0) → False</code>
	<code>list()</code>	tuple, str, dict	<code>list((1,2)) → [1,2]</code>
	<code>tuple()</code>	list, str	<code>tuple([1,2]) → (1,2)</code>
	<code>dict()</code>	iterable of pairs	<code>dict([('a',1)]) → {'a':1}</code>
Math	<code>abs()</code>	int, float, complex	<code>abs(-5) → 5</code>
	<code>round()</code>	int, float	<code>round(3.6) → 4</code>
	<code>pow()</code>	int, float	<code>pow(2,3) → 8</code>
	<code>max()</code>	iterable or multiple args	<code>max(1,5,3) → 5</code>
	<code>min()</code>	iterable or multiple args	<code>min([1,5,3]) → 1</code>
	<code>sum()</code>	iterable of numbers	<code>sum([1,2,3]) → 6</code>
Iterable	<code>len()</code>	str, list, tuple, dict	<code>len([1,2,3]) → 3</code>
	<code>range()</code>	int (end), optional start/step	<code>range(3) → 0,1,2</code>
	<code>enumerate()</code>	iterable	<code>list(enumerate(['a','b'])) → [(0,'a'), (1,'b')]</code>
	<code>sorted()</code>	iterable	<code>sorted([3,1]) → [1,3]</code>
	<code>reversed()</code>	iterable	<code>list(reversed([1,2,3])) → [3,2,1]</code>
Object Inspection	<code>type()</code>	any object	<code>type(5) → <class 'int'></code>
	<code>id()</code>	any object	<code>id(5)</code>
	<code>dir()</code>	any object	<code>dir([])</code>
	<code>help()</code>	any object	<code>help(len)</code>
Input/Output	<code>print()</code>	any type	<code>print("Hi")</code>
	<code>input()</code>	-	<code>name = input("Enter: ")</code>

Category	Function	Applies To	Example
Boolean / Conditional	<code>all()</code>	iterable of bools	<code>all([True,False]) → False</code>
	<code>any()</code>	iterable of bools	<code>any([True,False]) → True</code>
	<code>isinstance()</code>	any object + type	<code>isinstance(5,int) → True</code>
Utility	<code>chr()</code>	int (ASCII code)	<code>chr(65) → 'A'</code>
	<code>ord()</code>	str (single char)	<code>ord('A') → 65</code>

2. Dot Methods by Data Type

int

- **Dot Methods:** None (use operators like +, -, *, /)

float

- **Dot Methods:** None (use operators like +, -, *, /)

bool

- **Dot Methods:** None

tuple

- **Dot Methods:** `count(x)`, `index(x)`

str

Method	Description	Example
<code>capitalize()</code>	Capitalize first letter	<code>'hello'.capitalize() → 'Hello'</code>
<code>casefold()</code>	Lowercase for caseless comparison	<code>'ß'.casefold() → 'ss'</code>
<code>center(width)</code>	Center string in width	<code>'hi'.center(5) → ' hi '</code>
<code>count(sub)</code>	Count occurrences	<code>'hello'.count('l') → 2</code>
<code>encode()</code>	Encode to bytes	<code>'hi'.encode() → b'hi'</code>
<code>endswith(suffix)</code>	Check ending	<code>'test'.endswith('t') → True</code>
<code>find(sub)</code>	Find substring	<code>'hello'.find('l') → 2</code>
<code>format(*args, **kwargs)</code>	Format string	<code>'{} {}'.format('a','b') → 'a b'</code>
<code>index(sub)</code>	Find substring index	<code>'hello'.index('l') → 2</code>
<code>isalnum()</code>	Alphanumeric check	<code>'abc123'.isalnum() → True</code>
<code>isalpha()</code>	Alphabet check	<code>'abc'.isalpha() → True</code>
<code>isdigit()</code>	Digit check	<code>'123'.isdigit() → True</code>
<code>islower()</code>	Lowercase check	<code>'abc'.islower() → True</code>
<code>isspace()</code>	Whitespace check	<code>' '.isspace() → True</code>
<code>istitle()</code>	Titlecase check	<code>'Hello World'.istitle() → True</code>
<code>isupper()</code>	Uppercase check	<code>'ABC'.isupper() → True</code>
<code>join(iterable)</code>	Join iterable with string	<code>','.join(['a','b']) → 'a,b'</code>
<code>lower()</code>	Lowercase	<code>'HELLO'.lower() → 'hello'</code>
<code>lstrip()</code>	Strip left whitespace	<code>' hi'.lstrip() → 'hi'</code>
<code>replace(old,new)</code>	Replace substring	<code>'abc'.replace('a','x') → 'xbc'</code>
<code>rfind(sub)</code>	Rightmost find	<code>'hello'.rfind('l') → 3</code>
<code>rstrip()</code>	Strip right whitespace	<code>'hi '.rstrip() → 'hi'</code>
<code>split(sep)</code>	Split string	<code>'a,b'.split(',') → ['a','b']</code>
<code>startswith(prefix)</code>	Check prefix	<code>'hello'.startswith('h') → True</code>
<code>strip()</code>	Strip both sides	<code>' hi '.strip() → 'hi'</code>
<code>title()</code>	Titlecase string	<code>'hello world'.title() → 'Hello World'</code>
<code>upper()</code>	Uppercase	<code>'hi'.upper() → 'HI'</code>
<code>zfill(width)</code>	Pad with zeros	<code>'42'.zfill(5) → '00042'</code>

list

Method	Description	Example
<code>append(x)</code>	Add element	<code>[1,2].append(3) → [1,2,3]</code>
<code>clear()</code>	Remove all elements	<code>[1,2].clear() → []</code>
<code>copy()</code>	Shallow copy	<code>l2 = [1,2].copy()</code>
<code>count(x)</code>	Count occurrences	<code>[1,2,2].count(2) → 2</code>
<code>extend(iterable)</code>	Extend list	<code>[1].extend([2,3]) → [1,2,3]</code>
<code>index(x)</code>	Find index	<code>[1,2,3].index(2) → 1</code>
<code>insert(i,x)</code>	Insert at index	<code>[1,3].insert(1,2) → [1,2,3]</code>
<code>pop(i)</code>	Remove by index	<code>[1,2,3].pop(1) → 2</code>
<code>remove(x)</code>	Remove first occurrence	<code>[1,2,3,2].remove(2) → [1,3,2]</code>
<code>reverse()</code>	Reverse list	<code>[1,2,3].reverse() → [3,2,1]</code>
<code>sort()</code>	Sort list	<code>[3,1,2].sort() → [1,2,3]</code>

dict

Method	Description	Example
<code>clear()</code>	Remove all items	<code>d.clear()</code>
<code>copy()</code>	Shallow copy	<code>d2 = d.copy()</code>
<code>get(key, default)</code>	Get value	<code>d.get('a',0)</code>
<code>items()</code>	Key-value pairs	<code>d.items()</code>
<code>keys()</code>	Keys view	<code>d.keys()</code>
<code>pop(key)</code>	Remove key	<code>d.pop('a')</code>
<code>popitem()</code>	Remove last item	<code>d.popitem()</code>
<code>setdefault(key, default)</code>	Get or insert	<code>d.setdefault('a',0)</code>
<code>update(other)</code>	Merge dicts	<code>d.update({'b':2})</code>
<code>values()</code>	Values view	<code>d.values()</code>

Tips

- Methods like `len()`, `sorted()`, `list()` work on **any iterable**.
- `.copy()` → shallow copy; `import copy` → allows deep copies.