🐍

# Built-in Functions Cheat Sheet

Reference: https://docs.python.org/3/library/functions.html
Prepared by: Shehab Abdel-Salam

## Common Functions in Python

1. `abs(x)` Returns the absolute value of a number.

   ```
   print(abs(-5))  # Output: 5
   ```

2. `int(x)`: Converts a number or string to an integer.

   ```
   print(int(3.7))  # Output: 3
   print(int("10"))  # Output: 10
   ```

3. `float(x)`: Converts a number or string to a floating-point number.

   ```
   print(float(5))  # Output: 5.0
   print(float("3.14"))  # Output: 3.14
   ```

4. `len(s)`: Returns the length (number of items) of an object.

   ```
   print(len([1, 2, 3]))  # Output: 3
   print(len("hello"))  # Output: 5
   ```

5. `max(iterable, *[, key, default])`: Returns the largest item in an iterable or the largest of two or more arguments.

   ```
   print(max(1, 3, 2))  # Output: 3
   print(max([1, 2, 3]))  # Output: 3
   ```

6. `min(iterable, *[, key, default])`: Returns the smallest item in an iterable or the smallest of two or more arguments.

```
print(min(1, 3, 2))  # Output: 1
print(min([1, 2, 3]))  # Output: 1
```

7. `pow(x, y[, z])`: Returns x raised to the power y; if z is present, returns x raised to the power y, modulo z.

```
print(pow(2, 3))  # Output: 8
print(pow(2, 3, 3))  # Output: 2 (8 % 3)
```

8. `print(*objects, sep=' ', end='\n')`: Prints objects to the output, separated by sep and followed by end.

```
print("Hello world")
print("A", "B", sep=", ")  # Output: A, B
```

9. `round(number[, ndigits])`: Rounds a number to a given precision in decimal digits.

```
print(round(3.14159, 2))  # Output: 3.14
print(round(3.5))  # Output: 4
```

10. `str(object)`: Returns a string version of object.

```
print(str(123))  # Output: '123'
print(str(3.14))  # Output: '3.14'
```

11. `tuple(iterable)`: Converts an iterable to a tuple.

```
print(tuple([1, 2, 3]))  # Output: (1, 2, 3)
print(tuple("abc"))  # Output: ('a', 'b', 'c')
```

12. `list(iterable)`: Converts an iterable to a list.

```
print(list((1, 2, 3)))  # Output: [1, 2, 3]
print(list("abc"))  # Output: ['a', 'b', 'c']
```

13. `dict(**kwargs)`: Creates a new dictionary from keyword arguments.

```
print(dict(a=1, b=2))  # Output: {'a': 1, 'b': 2}
print(dict([('a', 1), ('b', 2)]))  # Output: {'a': 1, 'b': 2}
```

14. `set(iterable)`: Creates a new set object.

```
print(set([1, 2, 2, 3]))  # Output: {1, 2, 3}
print(set("hi"))  # Output: {'h', 'i'}
```

15. `sum(iterable, /, start=0)`: Sums the items of an iterable from left to right and returns the total.

```
print(sum([1, 2, 3]))  # Output: 6
print(sum([1, 2, 3], 10))  # Output: 16
```

16. `sorted(iterable, *, key=None, reverse=False)`: Returns a new sorted list from the items in an iterable.

```
print(sorted([3, 1, 2]))  # Output: [1, 2, 3]
print(sorted("bca"))  # Output: ['a', 'b', 'c']
```

17. `any(iterable)`: Returns `True` if any element of the iterable is true; otherwise, `False`.

```
print(any([0, 1, 2]))  # Output: True
print(any([0, 0, 0]))  # Output: False
```

18. `all(iterable)`: Returns `True` if all elements of the iterable are true; otherwise, `False`.

```
print(all([1, 2, 3]))  # Output: True
print(all([1, 0, 3]))  # Output: False
```

19. `zip(*iterables)`: Returns an iterator of tuples, where the i-th tuple contains the i-th element from each of the argument iterables.

```
print(list(zip([1, 2], ['a', 'b'])))  # Output: [(1, 'a'), (2, 'b')]
```

20. `map(function, iterable, ...)`: Applies a function to all the items in an input list.

```
print(list(map(str, [1, 2])))  # Output: ['1', '2']
```

21. `filter(function, iterable)`: Constructs an iterator from elements of iterable for which function returns true.

```
print(list(filter(lambda x: x % 2 == 0, [1, 2, 3, 4])))
# Output: [2, 4]
```