

# Function Arguments in Python & JavaScript

A complete cheat sheet explaining all argument types with examples.

## 1■■■ Positional Arguments

Python: Order of arguments matters.

```
def greet(name, age):  
    print(f"Hello {name}, you are {age} years old.")  
  
greet("Simon", 25)
```

JavaScript: Works the same way — based on order.

```
function greet(name, age) {  
    console.log(`Hello ${name}, you are ${age} years old.`);  
}  
greet("Simon", 25);
```

## 2■■■ Default Arguments

Python: Use default values when not provided.

```
def greet(name, age=18):  
    print(f"Hello {name}, you are {age} years old.")  
  
greet("Simon")          # Uses default  
greet("Ella", 25)       # Overrides default
```

JavaScript: Also supports default parameters.

```
function greet(name, age = 18) {  
    console.log(`Hello ${name}, you are ${age} years old.`);  
}  
greet("Simon");  
greet("Ella", 25);
```

## 3■■■ Arbitrary Positional Arguments

Python: Uses \*args to collect multiple values into a tuple.

```
def add_numbers(*numbers):  
    print(numbers)  
    print(sum(numbers))  
  
add_numbers(1, 2, 3, 4, 5)
```

JavaScript: Uses rest syntax (...args) to collect arguments into an array.

```
function addNumbers(...numbers) {  
    console.log(numbers);  
    console.log(numbers.reduce((a, b) => a + b));  
}  
  
addNumbers(1, 2, 3, 4, 5);
```

## 4■■■ Arbitrary Keyword Arguments

Python: Uses \*\*kwargs for named arguments.

```
def introduce(**person):  
    print(person)
```

```

    print(f"My name is {person['name']} and I am {person['age']} years old.")

introduce(name="Simon", age=25)

```

JavaScript: Passes keyword-style arguments as objects.

```

function introduce(person) {
    console.log(person);
    console.log(`My name is ${person.name} and I am ${person.age} years old.`);
}

introduce({ name: "Simon", age: 25 });

```

## 5 ■ Mixing Multiple Argument Types

Python: Combines \*args and \*\*kwargs.

```

def order_pizza(size, *toppings, **details):
    print(f"Size: {size}")
    print("Toppings:", toppings)
    print("Details:", details)

order_pizza("Large", "pepperoni", "cheese", address="Lagos", delivery=True)

```

JavaScript: Combines rest parameters and objects.

```

function orderPizza(size, ...toppingsAndDetails) {
    console.log("Size:", size);
    console.log("Other arguments:", toppingsAndDetails);
}

orderPizza("Large", "pepperoni", "cheese", { address: "Lagos", delivery: true });

```

## ■ Summary Comparison Table

Type	Python	JavaScript	Stored As
Positional	def f(a, b)	function f(a, b)	Variables
Default	def f(a=1)	function f(a=1)	Variables
Variable Positional	*args	...args	Tuple / Array
Variable Keyword	**kwargs	Object argument	Dict / Object
Keyword-only	def f(*, x)	N/A (use object)	N/A
Positional-only	def f(x, /)	N/A	N/A
Arbitrary args (legacy)	N/A	arguments	Array-like object