

# Python Function Parameters

---

## Textbook

---

## Python Function Parameters



Remember how to create a basic Python function?

```
1 def weather():  
2     print("It's a soggy day outside")  
3  
4  
5 weather()
```

Try it!

### Parameters

Now we will learn what goes inside the function parenthesis. [Function parameters](#) are what go inside the parentheses of the function declaration. These parameters can then be used inside the function.

```
1 def weather(type):  
2     print("It's a soggy day outside because it is " + type)  
3  
4  
5 weather()
```

## Arguments

Notice that the same word inside the parenthesis can be found inside the print statement, which is inside the function. Now, to determine what the parameter named forecast is, you say so when you call the function.

```
1 def weather(forecast):  
2     print("It's a soggy day outside because it is " + forecast)  
3  
4  
5 weather("raining")
```

Try it!

The information that goes inside the parameters when you call the function is called an [argument](#).



Notice that when we call the function named weather, we put in the string "raining". The function then takes that string and puts it inside the string using [concatenation](#). The value of `"raining"` is the [function's parameter](#).

See what happens when you call the function multiple times with different parameters.

```
1 def weather(forecast):
2     print("It's a soggy day outside because it is " + forecast)
3
4
5 weather("raining")
6 weather("snowing")
7 weather("drizzling")
```

Try it!

## The Return Statement

The return statement causes the function to stop executing.

```
1 def weather():
2     return "Hello"
3
4 print(weather())
```

Try it!

Functions that have a return statement are sometimes called "fruitful functions" because they give the programmer something they can use.

*Note: A return statement immediately quits the function. Notice how the following code doesn't print **How are you?***

```
1 def weather():
2     return "Hello"
3     print("How are you?")
4
5 print(weather())
```

Try it!

*Note: If a function does not have a return statement, it will automatically return **None***

## What data types can be passed as a parameter?

- Integers
- Strings
- Boolean values
- Floating point
- Lists
- Dictionaries, Tuples, Sets (You will learn more about these in later lessons)

## Documentation of Functions

Functions have some built in documentation that helps clarify what the function is doing. Let's take a look at some examples.

Here is a function like we have been practicing with.

```
1 def weather(first):
2     return first
3
4 print(weather("Snowy"))
```

Try it!

Now let's add some built-in documentation for the function.

```
1 def weather(first:str):
2     return first
3
4 print(weather("Snowy"))
```

Try it!

The `str` value after the parameter named `first` is simply saying that the data type for the parameter will be a string. This just helps to clarify what the function is doing.

Let's look at some further documentation that can go in a function.

```
1 def weather(first:str)-> str:
2     return first
3
4 print(weather("Snowy"))
```

Try it!

The `-> str` after the function declaration and parameters is saying what data type will be returned with the function. This function is returning a string.

Let's look at another example.

## Why Use Functions?

Breaking code down into smaller chunks such as functions really helps improve your code. Here are a few reasons why:

- **Maintainability** – Smaller, simpler functions are easier to maintain
- **Reusability** – Functions encourage code reuse instead of doing the same work over again
- **Debugability** – It's easier to debug several small functions than one large program

# Characteristics of a Well Defined Function

Here are some characteristics of a well-defined Python functions:

1. **Clear Purpose:** A good function has a clear job or purpose. Just like a well-written recipe, it should do one thing well, whether it's calculating a number, formatting text, or any specific task.
2. **Meaningful Name:** The name of the function should clearly describe what it does. This helps other programmers (and your future self!) understand its purpose without needing to dive into the details of its code.
3. **Input Parameters:** Functions often take inputs called parameters. These are like ingredients in a recipe. A well-defined function specifies what inputs it needs to work correctly.
4. **Output:** Functions usually produce some kind of output, just like a recipe produces a finished dish. It's important that the function clearly defines what it will return as its result.
5. **Modularity:** Functions promote modularity, meaning they can be used and reused in different parts of your program. This makes your code more organized and easier to maintain.
6. **Documentation:** Good functions often include comments or documentation that explain how to use them and what they do. This helps anyone using your function understand its purpose and how to interact with it.

## Checkpoint

---

### Python Function Parameters

1. Create a function that includes a **parameter**.
2. Inside the function include a print statement that prints out a string.
3. Concatenate the string with the parameter in the print statement.
4. Call the function.
5. Call the function **3 times** with different parameters.

### Requirements:

- Create a function that includes a parameter
- Inside the function include a print statement that prints out a string.
- Concatenate the string with the parameter in the print statement
- Call the function
- Call the function 3 times with different parameters.

## Questions (11)

---

### 1. What goes inside the parentheses of a function?

MULTIPLE CHOICE

Choose the correct answer:

- A. parameter
- B. value
- C. string
- D. property

## 2. True or False: You can call a function multiple times with different values for parameters?

MULTIPLE CHOICE

 Question image

Choose the correct answer:

- A. True
- B. False

## 3. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 def practice(activity):  
2     print("I like to practice " + piano)  
3  
4 practice("piano")
```

## 4. What value is being passed as the parameter in the following code?

MULTIPLE CHOICE

```
def practice(activity): print("I like to practice " + activity) practice("piano")
```

Choose the correct answer:

- A. def
- B. activity
- C. practice
- D. piano

## 5. What will the following code print out?

MULTIPLE CHOICE

```
def practice(activity): print("I like to practice " + activity) practice("piano")
```

Choose the correct answer:

- A. "piano"
- B. "I like to practice activity"
- C. "I like to practice "
- D. "I like to practice piano"



MULTIPLE CHOICE

**6. What will the following code print out?**

```
def practice(skill): print("Today I will practice " + skill) practice("piano") practice("soccer") practice("karate")
```

**Choose the correct answer:**

- A. "Today I will practice piano" "Today I will practice soccer" "Today I will practice karate"
- B. "Today I will practice piano"
- C. "Today I will practice soccer"
- D. "Today I will practice karate"

SELECT MULTIPLE

**7. What will the following code print out? Select all that apply.**

```
def practice(skill): print("Today I will practice " + skill) print("I enjoy practicing " + skill) practice("piano") practice("soccer") practice("karate")
```

**Select all that apply:**

- A. Today I will practice piano
- B. Today I will practice soccer
- C. I enjoy practicing soccer
- D. I enjoy practicing karate
- E. Today I will practice karate

DEBUG CODE

**8. Edit the text box below to debug (fix) the code:****Code to Debug:**

```
1 def artwork( ):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
```

DEBUG CODE

**9. Edit the text box below to debug (fix) the code:****Code to Debug:**

```
1 artwork(tool):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
5 artwork("pencil")
6 artwork("clay")
```

## 10. Edit the text box below to debug (fix) the code:

DEBUG CODE

### Code to Debug:

```
1 def artwork(tool):
2     print("When making art, I like to use " + tool)
3
4 artwork(paint)
```

## 11. Which of the following is used to declare (or create) a function in Python?

MULTIPLE CHOICE

### Choose the correct answer:

- A. def
- B. function
- C. fun
- D. funct

## Challenges (6)

### 1. Hungry for Apples 2

1. We have a problem. The user is 'still' hungry! This time though, we're going to build a function that takes an input outside the function, and then prints it.
2. Unlike last time, first you have to ask the user for a type of fruit, then you have to build a function and pass it in as a parameter.
3. After you do that, use that parameter in a print statement to print the name of the fruit the hungry user passed in three times.
4. Be sure to call the function afterwards!

Keep your eyes out for Hungry for Apples 1 and 3!

Example:

Input: `Apple`

Output: `AppleAppleApple`

Another example:

Input: `Watermelon`

Output: `WatermelonWatermelonWatermelon`



## 2. Count the Days in a Month Function

1. Write a function that takes in a number of a month (**1 - 12**) as an input.
2. Make sure the input is an **integer**.
3. This input will also be the **parameter**.
4. Your function should print the number of days in that month for the year 2021.

For example:

Input: **3**

Output: **31**

Another example:

Input: **11**

Output: **30**

**Note:** these are the number of days in each month in 2021:

January: 31, February: 28, March: 31, April: 30, May: 31, June: 30, July: 31, August: 31, September: 30, October: 31, November: 30, December: 31

## 3. Odd/Even Checking Function

1. Write a function named **odd\_even** that checks if a number is odd or even.
2. Your program should call the function with a parameter included.
3. If the parameter is odd, include a print statement that says **odd** .
4. If the parameter is even include a print statement that says **even** .
5. Call the parameter **3 times** with different values.

### Requirements:

- Create a function named **odd\_even**
- Include a parameter keyword in your function.
- Inside the function, check to see if the parameter is even. If it is, include a print statement in the function that says **even** .
- Inside the function, else, include a print statement in the function that says **odd** .
- Call the function named **odd\_even** 3 times with different values for parameters.

#### 4. A Party for the Letter C!

You decided that you want to throw a "C" party! You will have snacks that all start with c: cake, cheddar crackers, cheese, candy, caramel, etc. You will do activities that start with c: coloring, chalk, chat, create, etc.

1. Create a function that will determine if an input can be a part of your party.
2. The program will ask the user to **input** a snack or activity.
3. The input will then become the **parameter** for a function.
4. If the snack or activity starts with the letter "c" (lower or upper case), the function will print out, " `_____ can be a part of the party!` ".
5. If the input does not start with the letter "c", the function will print out, " `Sorry! _____ doesn't start with the letter c!` "

For example:

Input: `cake`

Output: `cake can be a part of the party!`

Another example:

Input: `carrots`

Output: `carrots can be part of the party!`

HINT: `my_list = list("candy")` will separate the string " `candy` " into individual letters.

**Don't forget to match the output exactly! Check punctuation, capitalization, spelling, and spacing.**

#### 5. Is it a Leap Year?

Write a function that receives a year as input (any year after 0).

This input will also be the parameter of the function.

If that year was a leap year, print `Leap year!` If not, print `Not a leap year.`

For example:

Input: `2020`

Output: `Leap year!`

Another example:

Input: `1991`

Output: `Not a leap year.`

**Hint:** A year was a leap year if it is evenly divisible by 4 but not divisible by 100.

**Don't forget to match the output exactly! Check punctuation, capitalization, spelling, and spacing.**

## 6. Area of a Circle

1. Create a function that will print out the area of a circle.
2. The user will **input** a radius.
3. This input will become the **parameter**.
4. The program will then print out the area of the circle.

**Reminder:** The area of a circle is  $3.14159 * \text{radius} * \text{radius}$ . Round your answer to one decimal place.

For example:

Input: 5

Output: 78.5

Another example:

Input: 2

Output: 12.6

**Hint:** to round use this code as an example:

```
round(numbertoround, 1)
```

The first space inside the parentheses is the number to be rounded, and the second space is the number of decimal places to be rounded to.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Python Function Parameters

```
1 def practice(activity):  
2     print("I like to practice " + activity)  
3  
4  
5 practice("piano")  
6 practice("soccer")  
7 practice("karate")
```

### Questions

#### 1. What goes inside the parentheses of a function?

MULTIPLE CHOICE

Correct Answer:

- |              |             |
|--------------|-------------|
| A. parameter | ✓ Correct   |
| B. value     | ✗ Incorrect |
| C. string    | ✗ Incorrect |
| D. property  | ✗ Incorrect |

#### Explanation:

These help make functions useful in more ways

#### 2. True or False: You can call a function multiple times with different values for parameters?

MULTIPLE CHOICE

Correct Answer:

- |          |             |
|----------|-------------|
| A. True  | ✓ Correct   |
| B. False | ✗ Incorrect |

#### Explanation:

You can call a function many times

### 3. Edit the text box below to debug (fix) the code:

DEBUG CODE

#### Incorrect Code:

```
1 def practice(activity):  
2     print("I like to practice " + piano)  
3  
4 practice("piano")
```

#### Correct Solution:

```
1 def practice(activity):  
2     print("I like to practice " + activity)  
3  
4 practice("piano")
```

#### Explanation:

The parameter name doesn't match its place inside the function.

### 4. What value is being passed as the parameter in the following code?

MULTIPLE CHOICE

#### Correct Answer:

- A. def ✗ Incorrect
- B. activity ✗ Incorrect
- C. practice ✗ Incorrect
- D. piano ✓ Correct

#### Explanation:

The parameter is the value inside the parentheses in the function declaration. The argument is in the function call.

### 5. What will the following code print out?

MULTIPLE CHOICE

#### Correct Answer:

- A. "piano" ✗ Incorrect

B. "I like to practice activity"

✗ Incorrect

C. "I like to practice "

✗ Incorrect

D. "I like to practice piano"

✓ Correct

### Explanation:

The string "piano" will go into the function where it says to go

## 6. What will the following code print out?

MULTIPLE CHOICE

### Correct Answer:

A. "Today I will practice piano" "Today I will practice soccer" "Today I will practice karate"

✓ Correct

B. "Today I will practice piano"

✗ Incorrect

C. "Today I will practice soccer"

✗ Incorrect

D. "Today I will practice karate"

✗ Incorrect

### Explanation:

The function is called 3 times so it will run 3 times.

## 7. What will the following code print out? Select all that apply.

SELECT MULTIPLE

### Correct Answers:

A. Today I will practice piano

✓ Correct

B. Today I will practice soccer

✓ Correct

C. I enjoy practicing soccer

✓ Correct

D. I enjoy practicing karate

✓ Correct

E. Today I will practice karate

✓ Correct

### Explanation:

The function that has 2 print statements is called 3 times.

[DEBUG CODE](#)

## 8. Edit the text box below to debug (fix) the code:

### Incorrect Code:

```
1 def artwork( ):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
```

### Correct Solution:

```
1 def artwork(tool):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
```

### Explanation:

This function is missing the word "tool" inside the parentheses.

[DEBUG CODE](#)

## 9. Edit the text box below to debug (fix) the code:

### Incorrect Code:

```
1 artwork(tool):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
5 artwork("pencil")
6 artwork("clay")
```

### Correct Solution:

```
1 def artwork(tool):
2     print("When making art, I like to use " + tool)
3
4 artwork("paint")
5 artwork("pencil")
6 artwork("clay")
```

### Explanation:

The function needs the word def

[DEBUG CODE](#)

## 10. Edit the text box below to debug (fix) the code:

### Incorrect Code:

```
1 def artwork(tool):
2     print("When making art, I like to use " + tool)
3
```



```
4 artwork(paint)
```

### Correct Solution:

```
1 def artwork(tool):  
2     print("When making art, I like to use " + tool)  
3  
4 artwork("paint")
```

### Explanation:

The string in the function call needs quotation marks

## 11. Which of the following is used to declare (or create) a function in Python?

MULTIPLE CHOICE

### Correct Answer:

- |             |             |
|-------------|-------------|
| A. def      | ✓ Correct   |
| B. function | ✗ Incorrect |
| C. fun      | ✗ Incorrect |
| D. funct    | ✗ Incorrect |

### Explanation:

Short for define.

## Challenges

### 1. Hungry for Apples 2

#### Solution:

```
1 fruit = input("Please enter the name of a fruit: ")  
2  
3 def hungryForApples(choice):  
4     print(choice + choice + choice)  
5  
6  
7 hungryForApples(fruit)
```

### 2. Count the Days in a Month Function

#### Solution:

```

1 month = int(input("Which month? 1-12"))
2
3 def invitation (choice) :
4     if choice == 4 or choice == 6 or choice == 9 or choice == 11:
5         print(30)
6     elif choice == 2:
7         print(28)
8     else:
9         print( 31)
10
11
12 invitation (month)

```

### 3. Odd/Even Checking Function

Solution:

```

1 def odd_even(num):
2     if num % 2 == 0:
3         print("even")
4     else:
5         print("odd")
6
7 odd_even(12)
8 odd_even(5)
9 odd_even(9)

```

### 4. A Party for the Letter C!

Solution:

```

1 thing=input('Input a snack or activity to see if it is allowed at the party.')
2
3 my_thing= list(thing)
4 def allowed(thing):
5     if my_thing[0]=='c' or my_thing[0]=='C':
6         print('{} can be a part of the party!'.format(thing))
7     elif my_thing[0]!='c' or my_thing[0]!='C':
8         print("Sorry! {} doesn't start with the letter c!".format(thing))
9
10
11 allowed(thing)

```

### 5. Is it a Leap Year?

Solution:

```

1 year = int(input("What year?"))

```

```
2
3 def leap_year(choice):
4     if choice % 4 == 0 and choice % 100 !=0:
5         print("Leap year!")
6     else:
7         print("Not a leap year.")
8
9
10
11 leap_year(year)
```

## 6. Area of a Circle

Solution:

```
1 answer = int(input("What is the radius of the circle?"))
2
3 def circle (radius) :
4     answer = 3.14159 * radius * radius
5     print(round(answer, 1))
6
7
8 circle(answer)
```