Defining a function

image

```
print("Peel of the bananas")
print("Add some milk to it")
print("Add some dates")
print("Add some dry fruits")
print("Top up with ice cream")
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
print("Peel of the bananas")
print("Add some milk to it")
print("Add some dates")
print("Add some dry fruits")
print("Top up with ice cream")
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
print("Peel of the bananas")
print("Add some milk to it")
print("Add some dates")
print("Add some dry fruits")
print("Top up with ice cream")
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
# Define a function

def banana_shake():  # defining the function
    # do some work
    print("Peel of the bananas")
    print("Add some milk to it")
    print("Add some dates")
    print("Add some dry fruits")
    print("Top up with ice cream")
```

```
# calling a function

banana_shake()
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
banana_shake()
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
banana_shake()
```

```
Peel of the bananas
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
# Quiz

mango_shake()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_4636/2794658734.py in <module>
----> 1 mango_shake()

NameError: name 'mango_shake' is not defined
```

```
def tea():
    print("Tea, chai chai")

tea()
tea()
tea()
for i in range(10):
    tea()
```

```
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
Tea, chai chai
```

Passing a parameter to the function

- Do you want to make different functions for different fruit shakes?

```
# fruit shake

def fruit_shake(fruit): # fruit is a parameter
    print("Peel of the", fruit)
    print("Add some milk to it")
    print("Add some dates")
    print("Add some dry fruits")
    print("Top up with ice cream")



fruit_shake()
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_4636/279707
4494.py in <module>
----> 1 fruit_shake()

TypeError: fruit_shake() missing 1 required positional argument:
'fruit'
```

```
# When calling a function always pass the value of params
# fruit_shake()

fruit_shake("Mango")
```

```
Peel of the Mango
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
fruit_shake("Cheeku")
```

```
Peel of the Cheeku
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
fruit_shake("Kiwi")
```

```
Peel of the Kiwi
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
fruit_shake("Cheeku")
```

```
Peel of the Cheeku
Add some milk to it
Add some dates
Add some dry fruits
Top up with ice cream
```

```
# intro

def intro(name):
    print("Hey my name is", name)
```

```
intro("Rahul")
```

```
Hey my name is Rahul
```

```
intro("Amit ji")
```

```
Hey my name is Amit ji
```

```
x = 5
```

```
print(x)
print(5)
```

```
5
5
```

```
nam = input()
```

```
intro(nam)
```

```
 Adrija ji
```

```
Hey my name is Adrija ji
```

```
intro(5)
```

```
Hey my name is 5
```

```
## propose
```

```
def propose(name):
    print("Hey I love you", name)
```

```
propose("Emma Watson")
```

```
Hey I love you Emma Watson
```

Multiple paramaters

- Introduce your family
- These arguments follow positions

```
def family_intro(father, mother, sibling):
    print("My father's name is", father)
    print("My mother's name is", mother)
    print("My sibling's name is", sibling)
```

```
# family_intro("Papa")
# This will give error
```

```
family_intro("Papa", "Mummy", "Brother")
```

```
My father's name is Papa
My mother's name is Mummy
My sibling's name is Brother
```

```
# Quiz
```

```
# Parameters are positional
```

```
family_intro("Sibling", "Father", "Mother")
```

```
My father's name is Sibling
My mother's name is Father
My sibling's name is Mother
```

Docstrings

- Should I have Kept some Documentation?
- Tell me something about yourself please...

```
# add them
```

```
# print?
```

```
def add(a, b):
    print(a + b)
```

```
# add("2" + 2)
```

```
# add?
```

```
# add docstring to your function

def add(a, b):

    """
    add: this functions adds two values given to it
    a, b: Give same/relatable kind of values corresponding to a and b
    """

    print(a + b)
```

```
add?
```

```
Signature: add(a, b)
Docstring:
add: this functions adds two values given to it
a, b: Give same/relatable kind of values corresponding to a and b
File:
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_4636/523124
507.py
Type:      function
```

```
add(2, 4)
```

```
6
```

```
# print?
```

Return a function

print function shows all the values that it prints but actually it doesnt give any value

```
# lets revisit print

print(24)
```

```
24
```

```
type(print())
```

```


NoneType
```

```
a = print(24)
```

```
24
```

```python
print(a)
```

```
None
```

```python
print(type(a))
```

```
<class 'NoneType'>
```

```python
x = add(2, 4)
```

```
6
```

```python
print(x, type(x))
```

```
None <class 'NoneType'>
```

```python
# return a value

def add(a, b):
    return a + b
```

```python
x = add(2, 3)
```

```python
print(x, type(x))
```

```
5 <class 'int'>
```

```python
# Can function flow go beyond return statement

def example():
    print("Before return")
    return "This is returned"
    print("After return")
```

```python
print(example())
```

```
Before return
This is returned
```

```python
n = example()
```

```
Before return
```

```
n
```

```
'This is returned'
```

```
# Quiz

def square(x):
    return x*x

z = square(3)
y = square(5)

print(z + y)
```

```
34
```

```
def add_2_nums_with_return(n1, n2):
    return n1 + n2

y = add_2_nums_with_return(5, 6)
print(y)
```

```
11
```

```
print(add_2_nums_with_return(5, 6))
```

```
11
```

```
23
```

```
23
```

Some inbuilt functions

```
# Absolute function

abs(-12.5)
```

```
12.5
```

```
abs(12.5)
```

```
12.5
```

```
# round
```

```
round(2.333)
```

```
2
```

```
round(233.43)
```

```
233
```

```
# Round of the value upto 2 decimal places
```

```
round(2.34353413, 2)
```

```
2.34
```

Fahrenheit to celsius

```
# c = (5/9) * (f-32)
```

```
def fahrenheit_to_celsius(f):
    # given this value of farenheit
    c = (5/9) * (f-32)
    return round(c, 2)
```

```
fahrenheit_to_celsius(98)
```

```
36.67
```

```
fahrenheit_to_celsius(32)
```

```
0.0
```

```
# Doubts
```