Python Challenges

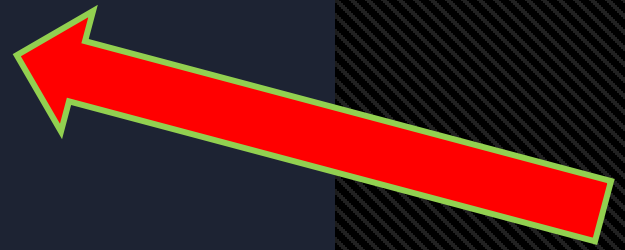
Problems to be solved using Python…

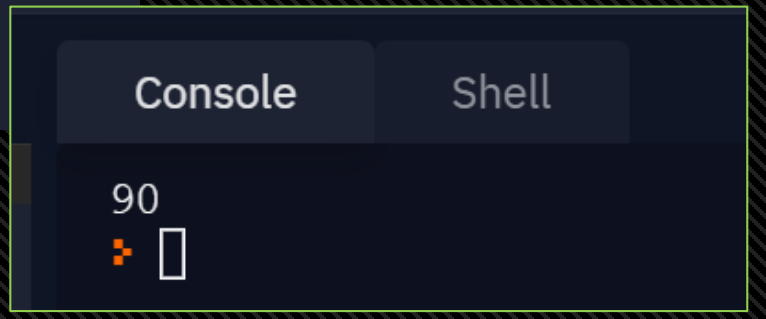
Using Functions

- Most of these will require you to define a function then return a value. You should then call the function. Like this…

Example...

```
1   #define the function
2   def MyranFun(a):
3       a   = a * 2
4       return a
5
6   #run/call the function
7   print(MyranFun(45))
```





Area of a Triangle

def Tri_area(b,a):

Difficulty: Very Easy

Challenge

Write a function that takes the base and height of a triangle and `return` its area.

Examples

```
tri_area(3, 2) → 3

tri_area(7, 4) → 14

tri_area(10, 10) → 50
```

Notes

- The area of a triangle is: `(base * height) / 2`
- Don't forget to `return` the result.
- If you get stuck on a challenge, find help in the **Resources** tab.
- If you're *really* stuck, unlock solutions in the **Solutions** tab.

Support

https://bitly.im/fA8en

https://bitly.im/IEY0W

Return the Next Number from the Integer Passed

def addition(num):

Difficulty: Very Easy

Challenge

Create a function that takes a number as an argument, increments the number by +1 and returns the result.

Examples

```
addition(0) → 1

addition(9) → 10

addition(-3) → -2
```

Notes

- Don't forget to `return` the result.
- If you get stuck on a challenge, find help in the **Resources** tab.
- If you're *really* stuck, unlock solutions in the **Solutions** tab.

Support

https://bitly.im/paJMD

https://bitly.im/MfSdu

Convert Age to Days

def addition(num):

Difficulty: Very Easy

Challenge

Create a function that takes the age and return the age in days.

Examples

```
calc_age(65) → 23725

calc_age(0) → 0

calc_age(20) → 7300
```

Notes

- Use **365 days** as the length of a year for this challenge.
- Ignore leap years and days between last birthday and now.
- Expect only positive integer inputs.

Support

https://bitly.im/T8dIY

Return a String as an Integer

def string_int(txt):

Difficulty: Very Easy

Challenge

Create a function that takes a string and returns it as an integer.

Examples

```
string_int("6") → 6

string_int("1000") → 1000

string_int("12") → 12
```

Notes

- All numbers will be whole.
- All numbers will be positive.

Support

https://bitly.im/1INqU

Find the Perimeter of a Rectangle

def find_perimeter(length, width):

Difficulty: Very Easy

Challenge

Create a function that takes `length` and `width` and finds the perimeter of a rectangle.

Examples

```
find_perimeter(6, 7) → 26

find_perimeter(20, 10) → 60

find_perimeter(2, 9) → 22
```

Notes

- Don't forget to `return` the result.
- If you're stuck, find help in the **Resources** tab.
- If you're really stuck, find solutions in the **Solutions** tab.

Support

mathopenref.com/rectangleperimeter.html

Sum of Polygon Angles

def sum_polygon(n):

Difficulty: Very Easy

Challenge

Given an n-sided regular polygon `n`, return the total **sum of internal angles** (in degrees).

Examples

```
sum_polygon(3) → 180

sum_polygon(4) → 360

sum_polygon(6) → 720
```

Notes

- `n` will always be greater than 2.
- The formula **(n - 2) x 180** gives the sum of all the measures of the angles of an **n-sided** polygon.

Support

- mathsisfun.com/geometry/interior-angles-polygons.html
- https://bitly.im/UHVkQ

Return the First Element in a List

def get_first_value(number_list):

Difficulty: Very Easy

Challenge

Create a function that takes a list containing only numbers and return the first element.

Examples

```
get_first_value([1, 2, 3]) → 1

get_first_value([80, 5, 100]) → 80

get_first_value([-500, 0, 50]) → -500
```

Notes

The first element in a list always has an index of 0.

Support

programiz.com/python-programming/list

Find Out the Leap Year

def leap_year(year):

Difficulty: Very Easy

Challenge

A leap year happens every four years, so it's a year that is perfectly divisible by four. However, if the year is a multiple of 100 (1800, 1900, etc), the year must be divisible by 400.

Write a function that determines if the year is a leap year or not.

Examples

```
leap_year(2020) → True

leap_year(2021) → False

leap_year(1968) → True
```

Support

Basic Variable Assignment

def name_string(name):

Difficulty: Very Easy

Challenge

A student learning Python was trying to make a function. His code should concatenate a passed string `name` with string `"Edabit"` and store it in a variable called `result`. He needs your help to fix this code.

Examples

```
name_string("Mubashir") → "MubashirEdabit"

name_string("Matt") → "MattEdabit"

name_string("python") → "pythonEdabit"
```

Support

```
def name_string(name):
    b == "Edabit"
    result == name + b
    return
```

Support

www.geeksforgeeks.org/python-list

The Farm Problem

def animals(chickens, cows, pigs):

Difficulty: Very Easy

Challenge

In this challenge, a farmer is asking you to tell him how many legs can be counted among all his animals. The farmer breeds three species:

- `chickens` = 2 legs
- `cows` = 4 legs
- `pigs` = 4 legs

The farmer has counted his animals and he gives you a subtotal for each species. You have to implement a function that returns the **total number of legs** of all the animals.

Examples

```
animals(2, 3, 5) → 36

animals(1, 2, 3) → 22

animals(5, 2, 8) → 50
```

Notes

- Don't forget to `return` the result.
- The order of animals passed is `animals(chickens, cows, pigs)`.
- Remember that the farmer wants to know the **total number of legs** and not the total number of animals.

Basketball Points

def points(twopointers, threepointers):

Difficulty: Very Easy

Challenge

You are counting points for a basketball game, given the amount of 3-pointers scored and 2-pointers scored, find the final points for the team and return that value ([2 -pointers scored, 3-pointers scored]).

Examples

```
points(1, 1) → 5

points(7, 5) → 29

points(38, 8) → 100
```

Support

www.programiz.com/python-programming/operators

Let's Fuel Up!

def calculate_fuel(num):

Difficulty: Very Easy

Challenge

A vehicle needs 10 times the amount of fuel than the distance it travels. However, it must always carry a minimum of 100 fuel before setting off.

Create a function which calculates the amount of fuel it needs, given the distance.

Examples

```
calculate_fuel(15) → 150

calculate_fuel(23.5) → 235

calculate_fuel(3) → 100
```

Notes

- Distance will be a number greater than zero.
- Return 100 if the calculated fuel turns out to be less than 100.

Support

https://bitly.im/ueLXr

Return Negative

def return_negative(n):

Difficulty: Very Easy

Challenge

Create a function that takes a number as an argument and returns negative of that number. Return negative numbers without any change.

Examples

```
return_negative(4) → -4

return_negative(15) → -15

return_negative(-4) → -4

return_negative(0) → 0
```

Support

www.w3schools.com/python/python_operators.asp

Make use of ABS in Python for this one!
https://www.programiz.com/python-programming/methods/built-in/abs

Are the Numbers Equal?

def is_same_num(num1, num2):

Difficulty: Very Easy

Challenge

Create a function that returns `True` when `num1` is equal to `num2`; otherwise return `False`.

Examples

```
is_same_num(4, 8) → False

is_same_num(2, 2) →  True

is_same_num(2, "2") → False
```

Notes

Don't forget to `return` the result.

Support

www.w3schools.com/python/python_conditions.asp