Intro to Programming

4-02-19

Sign In!

https://tinyurl.com/CADIntroPySp19-6

Last Week's Review

You are driving a little too fast, and a police officer stops you. Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

```
def caught_speeding(speed, is_birthday):
    # Your code here
```

```
caught_speeding(60, False) → 0
caught_speeding(65, False) → 1
caught_speeding(65, True) → 0
```

HINT: Use if conditions and returns!

Last Week’s Review Answer

```python
def caught_speeding(speed, is_birthday):
    if speed > 65 and speed <= 85 and is_birthday:
        return 1
    if speed >= 86 and is_birthday:
        return 2
    if speed <= 65 and is_birthday:
        return 0
    if speed > 60 and speed <= 80:
        return 1
    if speed >= 81:
        return 2
    return 0
```

Example of how to use functions

```
my_speed = 75
my_birthday = False

print("I am going " + my_speed + "mph.")
ticket = caught_speeding(my_speed, my_birthday)
if ticket:
    print("I was caught speeding.")
    print("My ticket was $" + 100 * ticket)
else:
    print("I was not caught speeding.")
```

Functions Review - Syntax

Defining your function:

```
def get_avg(parameter1, parameter2):
    '''Optional descriptive comment here'''
    # your code here
    # make use of parameter1 and parameter2
```

Calling your function:

```
function_name("Sean", [93, 65, 97])
```

Functions Review - Return Values

- Functions can also return some processed value with `return`
- Think of it as replacing that function call with the returned value

```
def is_passing(number):
    if number >= 70:
        return True
    return False
```

```
grade = 65
if is_even(grade):
    print(“Pass!”)
else:
    print(“Fail :(“)
def get_avg(grade_list):
    ’’’Returns the average of the list of grades.’’’
    totalSum = 0
    for grade in grade_list:
        totalSum += grade
    return totalSum/len(grade_list)

def grade_report(student, grade_list):
    ’’’Prints the grade report of a student.’’’
    avg = get_avg(grade_list)
    print(student, “‘s Grade Average: “, avg)

grade_report(“Sean”, [93, 65, 97])
```

```
def get_avg(grade_list):
    '''Returns the average of the list of grades.'''
    totalSum = 0
    for grade in grade_list:
        totalSum += grade
    return totalSum/len(grade_list)

def grade_report(student, grade_list):
    '''Prints the grade report of a student.'''
    avg = get_avg(grade_list)
    print(student, “‘s Grade Average: “, avg)

grade_report(“Sean”, [93, 65, 97]) Output: Sean ‘s Grade Average:  85.0
```

Keyword arguments

```python
def describe_pet(animal_type, pet_name):
    """ Display information about a pet. """
    print("I have a " + animal_type + ".")
    print("My " + animal_type + "'s name is " + pet_name.title() + ".")

describe_pet(animal_type="dog", pet_name="august")
describe_pet(pet_name="august", animal_type="dog")
```

```
Output:
I have a dog.
My dog's name is August.
I have a dog.
My dog's name is August.
```

- Keyword arguments are another way to call functions
  - They are name-value pairs you pass to a function
- The function is written the same, the only difference is in how you call it

Keyword arguments

```
def describe_pet(animal_type, pet_name):
    """ Display information about a pet. """
    print("I have a " + animal_type + ".")
    print("My " + animal_type + "'s name is " + pet_name.title() + ".")

describe_pet(animal_type="dog", pet_name="august")
describe_pet(pet_name="august", animal_type="dog")
```

```
Output:
I have a dog.
My dog's name is August.
I have a dog.
My dog's name is August.
```

- Keyword arguments are another way to call functions
    - They are name-value pairs you pass to a function
- The function is written the same, the only difference is in how you call it

Default values

```
def describe_pet(pet_name, animal_type="dog"):
    """ Display information about a pet. """
    print("I have a " + animal_type + ".")
    print("My " + animal_type + "'s name is " + pet_name.title() + ".")
describe_pet("august")
describe_pet("bevo", "cow")
```

Output

```
I have a dog.
My dog's name is August.
I have a cow.
My cow's name is Bevo.
```

- When writing a function you can specify a default value for each parameter which will be used if no arguments are specified in the function call
- Now if no `animal_type` is specified `‘dog’` will be used
- Parameters with default values must be listed after others for positional arguments to function correctly, otherwise Python will throw an error

Coding exercises

https://codingbat.com/python

Selected exercises:

- Logic-1: alarm_clock
- Warmup-2: string_bits
- List-2: count_evens

Thanks for coming!

- Next week - perhaps a project!
- Please fill out our feedback form, especially if you’d like to specify which topics we cover next week!
    - https://forms.gle/b9oJmWHhbhnpcaSB9