

# Heart Rate

## Task Goal

The goal of this assignment is for students to:

- write functions
- use arguments and parameters
- return values

To complete this task, students will also need to manipulate numeric types and work with Boolean values.

## Background

### HRmax

According to Wikipedia, **maximum heart rate (HRmax)** is "the highest heart rate an individual can achieve without severe problems through exercise stress, and generally decreases with age." Although it is difficult to accurately measure HRmax, there are many formulas for estimating it. One study compared many HRmax formulas for college-aged people and identified the Gellish2 equation, proposed by

researchers at Oakland University, as one of the most accurate for college-aged people.

## Target HR

According to Wikipedia, **Target Heart Rate (THR)**, is "a desired range of heart rate reached during aerobic exercise which enables one's heart and lungs to receive the most benefit from a workout." THR is dependent on HRmax. A common definition of THR is 65-85% of one's HRmax.

## Task

In a file named `heart_rate.py`, write two functions: **`gellish2`** and **`in_target_range`**

**`gellish2(age_min=1, age_max=100)`**

The `gellish2` function should take two default parameters `age_min` and `age_max` being assigned 1 and 100 as default values respectively. The function should include code to randomly select a number (for age) between `age_min` and `age_max` (**For this, you will need to import a module and call a function from the module that returns a random number when given a range**). The function

returns an estimated HRmax in beats per minute (bpm) based on the Gellish2 formula:

$$191.5 - (0.007 \times (\text{age}^2))$$

**The return type should be a float.**

**in\_target\_range(hr, age\_min=1, age\_max=100)**

The `in_target_range` function should take in three parameters, a non-default parameter for current heart rate (`hr`) and two default parameters `age_min` and `age_max` being assigned 1 and 100 as default values respectively. The function should return a Boolean value. It should return `True` if the heart rate (`hr`) is within (or exactly at a boundary of) the target range based on the given age, and `False` otherwise. You should define the target range to be from 65% to 85% of the HRmax (as estimated by the Gellish2 formula).

### Notes:

- **You should call your implementation of `gellish2` from inside of `in_target_range`. Don't put the formula in both functions.**

- **Your program should contain only the functions. No part of your program, including the functions, should either print or ask for any input.**
- **You should not have any code outside of your functions. Having code outside of your functions may cause all test cases to fail.**
- **Remember to include your name and ID as a comment at the top of the file.**
- **You should not use conditionals or any material not covered in lectures in your submission**
- **Comparison statements such as `x < 3` evaluate to boolean (bool) values in Python. See the [Troubleshooting](#) section, or section 6.1 of the textbook for more. True and False are not the same as the strings "True" and "False".**

## **Example Invocations**

When you run `heart_rate.py`, nothing should happen. It defines functions, it does not run them.

**`import heart_rate`**

If in another file (which you do not submit) you write the following:

**print(heart\_rate.gellish2())** *#assuming the randomly selected number for age is 18*

**print(heart\_rate.gellish2(29,29))** *#forcing the randomly selected number for age to be 29*

**print(heart\_rate.in\_target\_range(112))** *#assuming the randomly selected number for age is 95*

**print(heart\_rate.in\_target\_range(123,21,21))** *#forcing the randomly selected number for age to be 21*

**print(type(heart\_rate.in\_target\_range(123, 21,21)))** *#forcing the randomly selected number for age to be 21*

you should get the following output:

```
189.232
185.613
False
True
<class 'bool'>
```

## Troubleshooting

You should probably write *two* python files for this assignment; the one you turn in (`heart_rate.py`) that *defines* `gellish2` and `in_target_range`, and a separate file you use to test it.

In your testing file,

- Don't forget to import `heart_rate`
- Try both the examples we give above and other examples of your own.
- When adding examples, make sure you know what the right answer is supposed to be! How could you return a boolean value? Consider the following example:

```
def example(x, y):
```

```
    """returns True if x is bigger than y, False if it is not"""
```

```
    return x > y
```

Remember that Python cares about indentation, and that every more-indented block must be preceded by a colon :

Want to write something more than just `<` and `>`? There is a chart of other comparison operators (the Python equivalents of things like  $\neq$  and  $\leq$  in Python) in §6.1.2.

