National Centre for Computing Education

Raspberry Pi

# Order matters

## Task

### Step 1

Open the Python program below (ncce.io/py-order-20) in your development environment. The program is supposed to convert a length of time from seconds to minutes.

```
1  print(secs,"seconds is",mins,"whole minutes")
2  mins = secs // 60
3  secs = 2567
```

### Step 2

There are two variables in the program: `secs` and `mins`.
Some of the statements **assign** a value to a variable.
Some of the statements **refer to** (use) the value of a variable.

For each of the sentences below, write down which program statement(s) it applies to:
(fill in with 1, 2, 3, or any combination of these line numbers)

| | |
|---|---|
| | assigns a value to the `secs` variable |
| | assigns a value to the `mins` variable |
| | refers to the value of the `secs` variable |
| | refers to the value of the `mins` variable |

### Step 3

**Run** the program. You will be faced with an error message on line 1:

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

The statement on line 1 is executed first, and that statement refers to the `secs` variable.

However, no value has been assigned to `secs`, hence the `NameError`.

**Rearrange** (change the order of) the statements, so that the program runs to completion and displays the following message:

```
2567 seconds is 42 whole minutes
```

There are two things that you will need to remember:

- Program statements are executed in sequence, one after the other.
- When a variable is **referred to** in a statement, it must have previously been **assigned** a value, otherwise a `NameError` will arise.

National Centre for Computing Education

Raspberry Pi

# How to input numbers

## Worked example Age calculation

This Python program is supposed to compute and display the age of the user, given their year of birth.

```
1  print("Year of birth?")
2  birth_year = input()
3  age = 2020 - birth_year
4  print("You are", age, "years old")
```

If you **run** the program and type in your year of birth when you are prompted to, you will be faced with an **error message** on line 3:

```
age = 2020 - birth_year
TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

This is because **`input`** returns what the user has typed as a string, i.e. a piece of **text**. The value of **`birth_year`** is a piece of text, so `2020 - birth_year` cannot be evaluated.

### Step 1

**Modify** line 2. This is how user input is converted to an **`int`**eger value:

```
2  birth_year = int(input())
```

### Step 2

**Run** your program.

If you were faced with an **error message**, these are some of the common errors that may be responsible:

☐ missing one or both of **`int`**'s brackets misplacing one of `int`’s brackets

## Task 1 Your weight on the moon

Your science teacher asks you to make a program that reads the user’s weight on Earth and calculates how much the user will weigh on the moon.

You do some research and find out that gravity on the moon is a sixth (⅙) of what it is on Earth.

**Example**

**Note:** Use these numbers to test that your program works correctly. In general, the result displayed depends on user input, so it will not always be the same.

| | |
|---|---|
| The program displays a prompt and waits for keyboard input. | `Weight on Earth?` |
| The user types in a reply. | `60` |
| The program displays the result. | `Weight on moon: 10.0` |

### Step 1

Open this incomplete program (ncce.io/py-moon-20) in your development environment:

```
1  print("Weight on Earth?")
2  weight_earth = 
3  weight_moon  = 
4  print("Weight on moon:", weight_moon)
```

### Step 2

**Complete** line 2 so that the program receives input from the keyboard, after displaying a prompt to the user. Make sure that the value assigned to the `weight_earth` variable is an integer.

### Step 3

**Complete** line 3 so that the program calculates the weight on the moon to be one sixth (⅙) of the weight on Earth, i.e. one sixth of the value of the `weight_earth` variable.

## Task 2 Your age in dog years

You are going to make a program that reads the user's age and calculates how old the user is in dog years. The common perception is that a human year is equal to 7 dog years.

**Example**

**Note:** Use these numbers to test that your program works correctly. In general, the result displayed depends on user input, so it will not always be the same.

| | |
|---|---|
| The program displays a prompt and waits for keyboard input. | `How old are you?` |
| The user types in a reply. | `5` |
| The program displays the result. | `You are 35 years old in dog years` |

### Step 1

Write your program, run it, and test it. Use the code from the worked example and the previous task as a point of reference.

### Tip

You will need to use:

| | |
|---|---|
| `print` | for displaying messages to the user |
| `input` | for receiving keyboard input |
| `int` | for converting values to integers (whenever possible) |
| `=` | for performing assignments of expression values to variables |
| `*` | for multiplication |

## Explorer task for the worked example

The worked example will only work correctly as long as the current year is 2020. After that, it will need to be updated accordingly. You can modify the program so that it **knows** what the current year is.

### Step 1

Add these two lines of code to the beginning of the program:

```
1  from time import localtime
2  year = localtime().tm_year
```

### Explainer

Line 1 declares that the program will use a function called `localtime`, from the `time` module. Modules are **libraries** of code that we can use in our programs.

Line 2 invokes `localtime` to retrieve the current year.

You can use `localtime` to obtain any part of the current date and time, including what weekday it is (as an integer). Read the relevant documentation to find out more.

### Step 2

Replace any occurence of `2020` in your program with a reference to the variable `year`. Its value will **always** be the current year.

## Explorer task for the worked example

**Modify** the worked example so that it computes the user's age in days (approximately).

## Explorer task for 'Your weight on the moon'

A person's weight on the moon is 16.5% of what it is on Earth.

**Complete** line 3 so that the program calculates the user's weight on the moon according to this alternative description.

You can then run both versions of the program, input the same numbers, and compare the results to see if they are significantly different.

National Centre for Computing Education

Raspberry Pi

# Homework

## Task 1

Read the Python program below:

```
1  num1 = int(input())
2  num2 = 10 + num1 * 2
3  print(num2)
4  num1 = 20
5  print(num1)
```

### Question 1

When this program is executed, if the user types `10` on the keyboard, what will be displayed on the screen as a result of executing **line 3**?

A. `30`

B. `40`

C. `10 + 10 * 2`

D. `10 + num1 * 2`

### Question 2

When this program is executed, if the user types `10` on the keyboard, what will be displayed on the screen as a result of executing **line 5**?

A. `10`

B. `20`

C. `10` and `20`

D. There is an error in the program because a variable cannot hold two values at the same time

## Task 2

**Rearrange** the lines in the Python program below, so that you have a runnable program that holds a meaningful interaction with the user.

```
print("And where do you live", name)
print("I’ve never been to", location)
name = input()
print("What is your name?")
location = input()
```

Write your rearranged program below:

## Task 3

The **incomplete** program below is supposed to prompt the user for a distance in miles, and convert the distance that the user enters to kilometres.

```
1  print("Enter a distance in miles:")
2  miles = 
3  kilometres = 
4  print(miles, "miles is", kilometres, "km")
```

### Step 1

**Complete** line 2 so that the value assigned to the **`miles`** variable is obtained from what the user types on the keyboard.

## Step 2

**Complete** line 3 so that the program calculates the value of the `kilometres` variable to be the equivalent of the `miles` variable, converted to kilometres. Note that 1 mile is equal to 1.60934 kilometres.