



# Recapping Year 8 Python

Hillview International School Year 9

# Today's lesson

- Creating Python files
- Inputting, assigning and displaying values
- Functions
- Commenting
- If / elif / else statements
- Boolean operations and modifiers (==, and)
- Import

# Creating Python code files

- Create a new code file: **File -> New**
- A new window appears – this is a Code File. We'll write our program in this file.
- Save the file as **recap.py**
- **NOTE: You cannot run .py files by double-clicking them!**

# Inputting, assigning and displaying values

In your new file, have a go at entering these values and displaying them:

- Strings: 'Hello', "Goodbye"

To input a string, use:

```
my_string = input()
```

- Integers: 2, 6, -89, 45421

To input an integer at the screen, use:

```
my_number = int(input())
```

- Floats: 2.4, 6.0, -89.54, 111.111

To input a float number at the screen, use:

```
my_number = float(input())
```

- To display a variable:

```
print("That value was", my_number)
```

You have 3 minutes  
Run it when you're finished

www.Bandicam.com  
**00:03:00**  
000

# Putting functions in your code

Write the following code into a new file called `user_info.py`

```
def process_user():  
    name = input("Please enter your name: ")  
    colour = input("What is your favourite colour? ")  
    birth_country = input("Where were you born? ")  
    shoe_size = input("What is your shoe size? ")  
  
    print("Hi", name + "!")  
    print("Your favourite colour is", colour + ". I like that too!")  
    print("You were born in", birth_country + ". I've never been there...")  
    print("You wear size", shoe_size, "shoes.")
```

Indentation tells Python that this code is part of the function definition

Function definition

`process_user()` ← Function call

- Run it when you're finished
- You have 5 minutes



# Commenting and spacing

```
# Function: Collect name and favourites then display them on the screen
def process_details():

    # Retrieve user's name and greet them
    name = input("What is your name? ")
    print("Hi,", name + "!")

    # Retrieve user's favourite number and display it
    number = input("What is your favourite number? ")
    print("Your favourite number is", number)

    # Retrieve user's favourite sports team and display it (use title case)
    team = input("What is your favourite sports team? ")
    print("Your favourite sports team is", team.title())

# Collect name and favourites then display them on the screen
process_details()
```

Now comment your code – you have 3 minutes

03:00

# If / elif / else statements

```
# Function: evaluate whether the student passed or failed
def pass_or_fail():
    mark = int(input("Please enter your mark: "))

    if mark >= 90:
        print("Grade A")
    elif mark >= 80:
        print("Grade B")
    elif mark >= 70:
        print("Grade C")
    elif mark >= 60:
        print("Grade D")
    elif mark >= 50:
        print("Grade E")
    else:
        print("Fail")

    print("Thank you")

# End function

#####

# Execute program
pass_or_fail()
```

- [Copy this code](#)
- [Run it when you're finished](#)
- You have 5 minutes



5:00

# Combining conditions with **and**

- We will now be using the **and** keyword.
- This allows us to add multiple conditions together
  - In maths you could do:  
 $50 < x < 70$
  - In programming, you do:  
`x > 50 and x < 70`



- Amend your function to look like this:

```
def pass_or_fail():  
    mark = int(input("Please enter your mark: "))  
    will_resit = input("Will you resit if you fail? (yes/no): ")  
  
    if mark >= 90:  
        print("Grade A")  
    elif mark >= 80:  
        print("Grade B")  
    elif mark >= 70:  
        print("Grade C")  
    elif mark >= 60:  
        print("Grade D")  
    elif mark >= 50:  
        print("Grade E")  
    elif mark < 50 and will_resit == "yes":  
        print("Please register to resit the exam.")  
    else:  
        print("Fail")  
  
    print("Thank you")
```



Add these lines of code



# `import` modules

- Python contains lots of built-in functions
- So far we've used `input()` and `print()`
- These are functions in the "core library" (built-in functions)
- There are more libraries, or "modules", that contain functions for certain tasks, such as networking and graphics
- The two modules we'll look at today are:
  - `time` – time related functions (in the task code only)
  - `random` – random number generators

# random module

- **random** includes functions related to random number generation
- The only function we'll use for now is **random.randint(x, y)**
- This returns a random integer in the range x to y (inclusive)
- Enable it by importing the time module:

```
import random
```

- Call the function like this:

```
random.randint(1,10)
```

```
# Random integer  
# from 1 to 10
```

# Task: Rolling the dice

- Copy the following code and see what happens:
- Save it as `dice.py`
- [You have 3 minutes](#)

```
import time
import random

def main():
    print("Let's roll a dice!")
    time.sleep(1)
    print("*shake*")
    time.sleep(1)
    print("*shake*")
    time.sleep(1)
    print("*roll!*")
    time.sleep(1)
    dice_value = random.randint(1,6)
    print("You rolled a", str(dice_value) + "!")

# Execute program
main()
```



# Rest of lesson and homework

- Text Adventure Game: Advanced version
- Due: Monday 18 September
- Include at least 8 rooms (including the existing rooms)
- Include at least 2 challenges (including the existing challenge)

# Next week

- Monday: advanced functions
  - Functions with parameters (recap)
  - Functions with return values
- Wednesday: advanced control
  - While loops
  - Nested if statements