



Hillview International School: Year 9

Programming in Python: Advanced Functions

Lesson outcomes

By the end of this lesson, you will be able to:

- Write and use functions that take parameters
- Write and use functions that return values
- Write and use recursive functions

Functions with parameters

- Allows functions to do different things depending on the parameters

- One you use already:

```
print(message)
```

- Defined like this:

```
def print_details(name, age):  
    print("Hi there,", name)  
    print("You are", age, "years-old!")
```

Parameters and arguments

- Parameters are variables in the function definition:

```
def print_name(name, age):
```

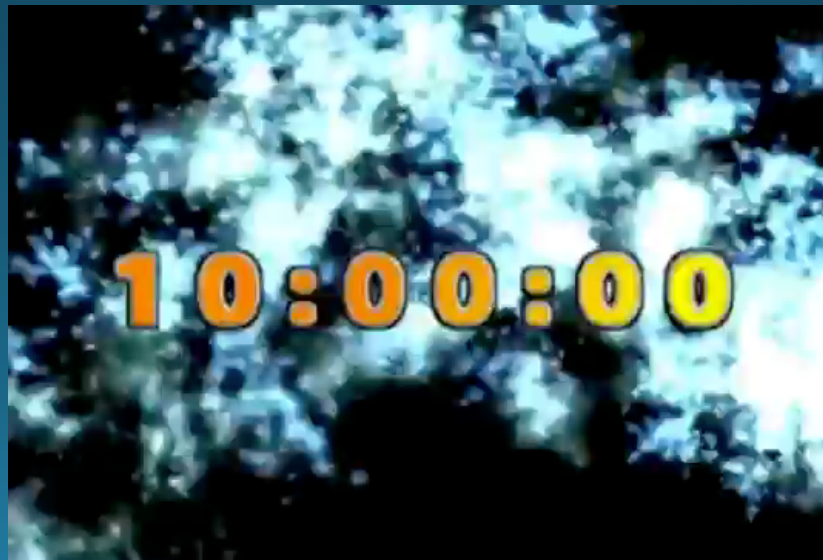
- Arguments are the variables passed into the function call:

```
print_name("Arthur", 86)
```

- Not unusual to see the words used interchangeably

Activity 1: User details

- The user enters their name, age and favourite colour
- The program prints a greeting including the user's name, age and favourite colour.
- Use: function with parameters to print the greeting.
- Time: you have 10 minutes!



Functions with return values

- Return values allow you to get answers from functions

- One you use already:

```
name = input("Please enter your name > ")
```

- Used like this:

```
def get_name():  
    name = input("Please enter your name")  
    return name
```

Activity 2: Get user's full name

- The user enters their first name, then their last name (at separate prompts).
- The function returns the first name and last name as one string then the program prints a greeting to the user by their full name.
- Use: a function with return value to get the user's full name.
- **Time:** you have 10 minutes!



09:58

Recursive functions

- A function that calls itself
- Used like this:

```
def my_func(x):  
    if x > 0:  
        my_func(x-1)  
# end my_func
```

Example: Factorials (intro)

- The factorial of a positive integer n , denoted by $n!$, is the product of all positive integers less than or equal to n .

For example:

$$3! = 3 \times 2 \times 1 = 6$$

- What is the value of $4!$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

- What is the value of $5!$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Example: Factorials (intro)

- Alternatively, $n!$ can be expressed as $n \times (n-1)!$
- For example:

$$5! = 5 \times 4!$$

$$4! = 4 \times 3!$$

$$3! = 3 \times 2!$$

$$2! = 2 \times 1!$$

$$1! = 1$$

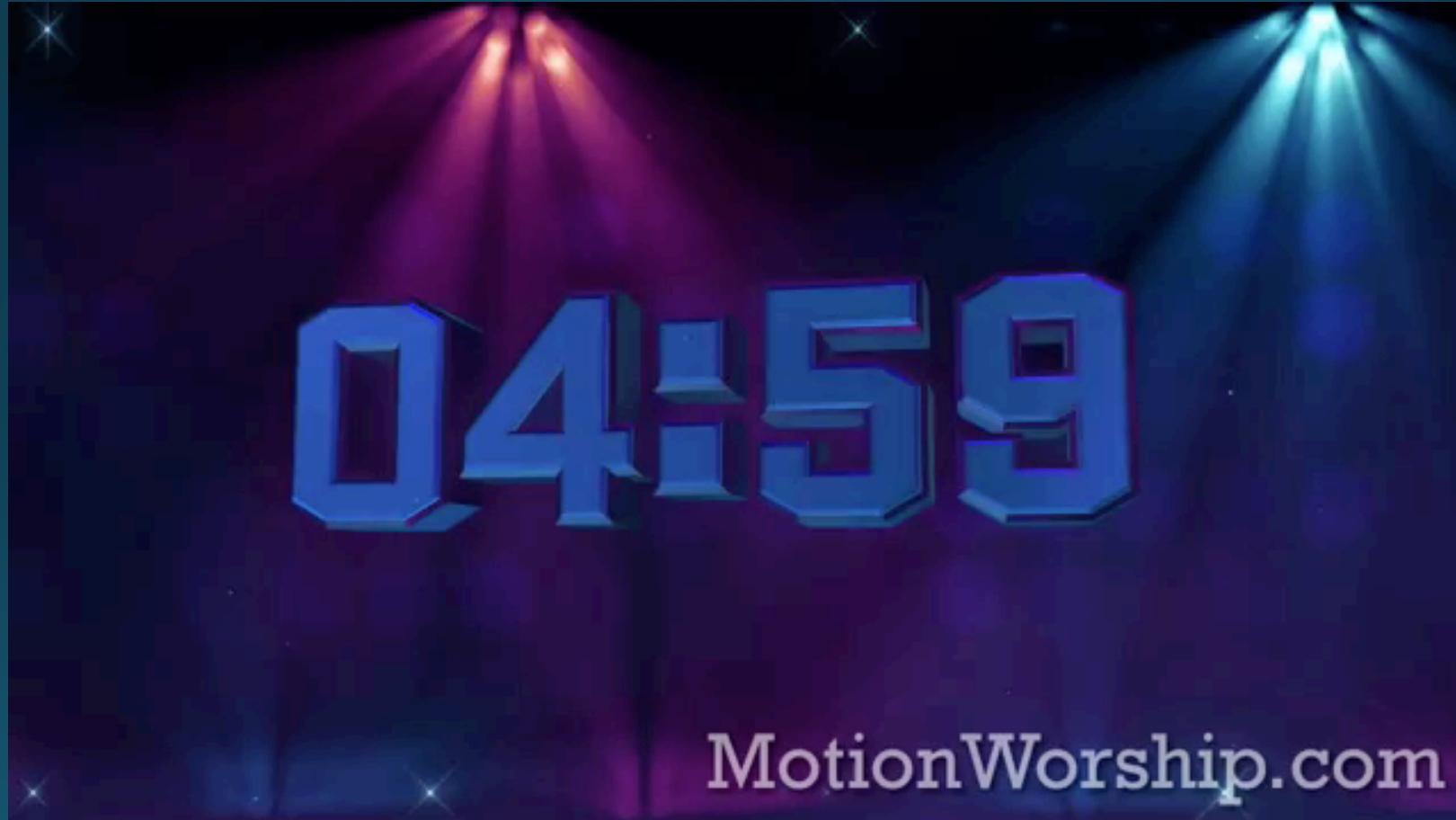
Example: Factorials

Write a program using a recursive function to calculate factorials

- Inputs: a positive integer (n)
- Outputs: the result of n!

```
def factorial(n):  
    if n == 1:  
        return 1  
    else:  
        return n * factorial(n-1)  
# end factorial function  
  
n = int(input("Choose a number to compute the factorial: "))  
print(str(n)+"! =", str(factorial(n)))
```


Activity 3: Copy the factorial program



Activity 4: Finding the square number

- The user enters a positive integer
- The program prints the largest square number smaller than or equal to the number entered.
- Use: recursive functions that take parameters and return values
- Example:
 - The user enters the number 32
 - The program prints that the largest square number under 32 is 25
- Extension: also print which number is the square root of this square number
- Time: you have the rest of the lesson plus homework time

Lesson summary

You should now be able to:

- Write and use functions that take parameters
- Write and use functions that return values
- Write and use recursive functions

On Wednesday:

- Lists
 - Arrays
 - Tuples
 - Associative arrays