

# Lambda Functions

# Lambda functions

Name comes from lambda calculus

Similar to arrow functions in JavaScript

Also known as anonymous functions, technically do not have a name

Often used in special situations where a function does not need a name,  
nor to be held in memory for later use

# Lambda function syntax

```
lambda arg1, arg2: expression to return
```

Argument list is comma-separated, not surrounded by parentheses

Multiple arguments allowed

Only one expression/line allowed in function body

Major difference from JavaScript arrow functions which allow multiline function bodies

# Lambda function example

```
lambda num: num ** 2
```

```
def square(num):  
    return num ** 2
```



# High-order functions & callback functions

Functions can be used as arguments for other functions

Higher-order functions are functions that use other functions as arguments

Callback functions are functions used by higher order functions as arguments

Lambda functions are typically used as callback functions

# Lambda function callback example

```
# This higher-order function has a parameter of a callback function
def a_function(callback):
    print(callback(3))
```

```
# This calls the higher-order function with a lambda function argument
a_function(lambda num : num ** 2) # this would print 9
```