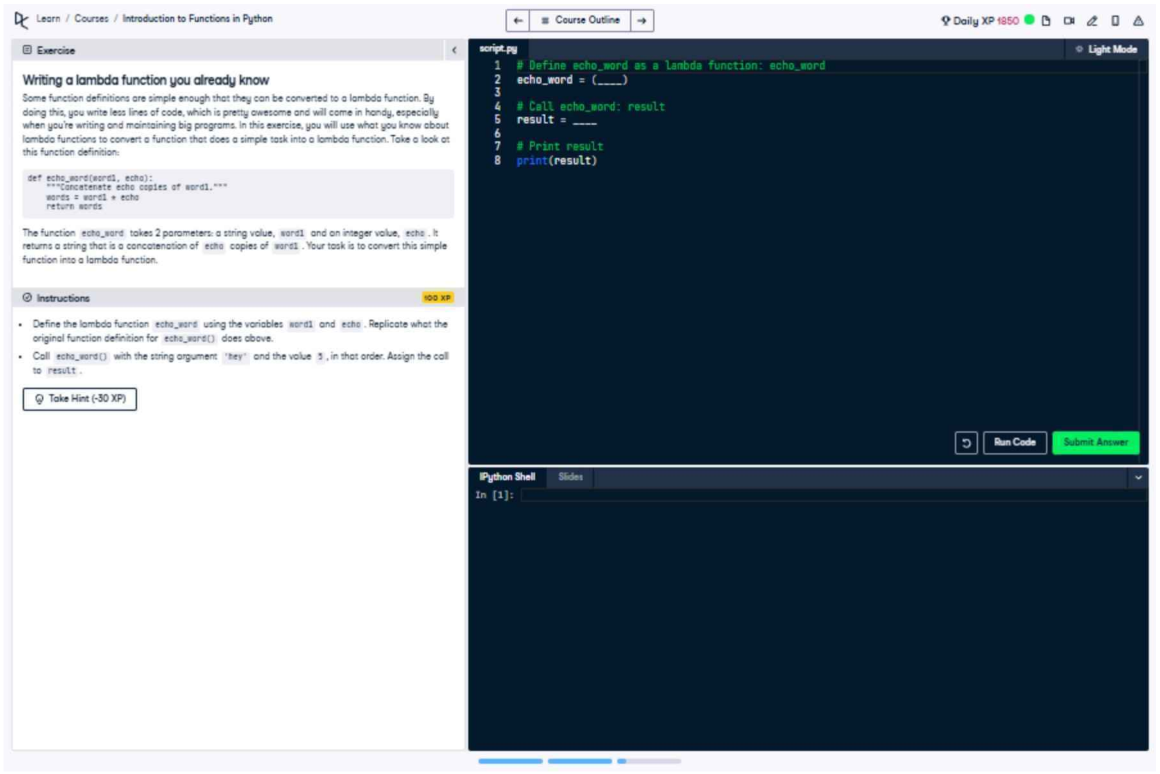
Writing a Lambda Function You Already Know



Writing a lambda function you already know

Some function definitions are simple enough that they can be converted to a lambda function. By doing this, you write less lines of code, which is pretty awesome and will come in handy, especially when you're writing and maintaining big programs. In this exercise, you will use what you know about lambda functions to convert a function that does a simple task into a lambda function. Take a look at this function definition:

```
def echo_word(word1, echo):
    """Concatenate echo copies of word1."""
    words = word1 * echo
    return words
```

The function `echo_word` takes 2 parameters: a string value, `word1` and an integer value, `echo`. It returns a string that is a concatenation of `echo` copies of `word1`. Your task is to convert this simple function into a lambda function.

Instructions 100 XP

- Define the lambda function `echo_word` using the variables `word1` and `echo`. Replicate what the original function definition for `echo_word()` does above.
- Call `echo_word()` with the string argument `'hey'` and the value `5`, in that order. Assign the call to `result`.

```
# Define echo_word as a lambda function: echo_word
echo_word = (____)

# Call echo_word: result
result = ____

# Print result
print(result)
```

```
In [1]:
```

Question:

Some function definitions are simple enough that they can be converted to a lambda function.
By doing this, you write less lines of code, which is pretty awesome and will come in handy,
especially when you’re writing and maintaining big programs.
In this exercise, you will use what you know about lambda functions to convert a function
that does a simple task into a lambda function.

Take a look at this function definition:

```
def echo_word(word1, echo):
    """Concatenate echo copies of word1."""
    words = word1 * echo
    return words
```

The function `echo_word` takes 2 parameters: a string value, `word1` and an integer value, `echo`.

It returns a string that is a concatenation of `echo` copies of `word1`. Your task is to convert this simple function into a lambda function.

Instructions:
1. Define the lambda function `echo_word` using the variables `word1` and `echo`. Replicate what the original function definition for `echo_word()` does above.
2. Call `echo_word()` with the string argument `'hey'` and the value `5`, in that order. Assign the call to `result`.
3. Print `result`.

Answer:

```
# Define echo_word as a lambda function: echo_word
echo_word = lambda word1, echo: word1 * echo

# Call echo_word: result
result = echo_word("hey", 5)

# Print result
print(result)
```

Explanation:

1. Lambda functions in Python are defined using the `lambda` keyword.
   The syntax is: `lambda arguments: expression`.

2. The original `echo_word` function concatenates `echo` copies of the string `word1`.
   In the lambda version, the function is compactly written as: `lambda word1, echo: word1 * echo`.

3. The `*` operator repeats the string `word1` exactly `echo` times.

4. We call the lambda function with `"hey"` as `word1` and `5` as `echo`.
   The result of `echo_word("hey", 5)` is `"heyheyheyheyhey"`, which is assigned to `result`.

5. Finally, we print the value of `result` to verify the output.