

Greater and Less Than

The screenshot shows a web browser with several tabs open. The active tab is 'Greater and less than | Python'. The browser address bar shows the URL: campus.datacamp.com/courses/intermediate-python/logic-control-flow-and-filtering/ex=3. The page content includes an exercise titled 'Greater and less than' with instructions and a code editor.

Exercise: Greater and less than

In the video, Hugo also talked about the less than and greater than signs, `<` and `>`. In Python, you can combine them with an equals sign: `<=` and `>=`. Pay attention: `<=` is valid syntax, but `<=<` is not.

All Python expressions in the following code chunk evaluate to `True`:

```
3 < 4
3 <= 4
"alpha" <= "beta"
```

Remember that for string comparison, Python determines the relationship based on alphabetical order.

Instructions (100 XP)

- Write Python expressions, wrapped in a `print()` function, to check whether:
 - `x` is greater than or equal to `-10`. (`x` has already been defined for you.)
 - `"test"` is less than or equal to `y`. (`y` has already been defined for you.)
 - `True` is greater than `False`.

Code Editor:

```
1 # Comparison of integers
2 x = -3 * 6
3
4
5 # Comparison of strings
6 y = "test"
7
8
9 # Comparison of booleans
10
```

Python Shell:

```
In [1]:
```

Below is the exercise on 'Greater and Less Than' from the Python course. The image includes the instructions, code, and questions to be answered.

Instructions:

Write Python expressions, wrapped in a `print()` function, to check whether:

1. `x` is greater than or equal to `-10`. (`x` has already been defined for you.)
2. `"test"` is less than or equal to `y`. (`y` has already been defined for you.)
3. `True` is greater than `False`.

Solution:

```
# Comparison of integers
```

```
x = -3 * 6
```

```
print(x >= -10) # Checks if x is greater than or equal to -10
```

```
# Comparison of strings
```

```
y = "test"
```

```
print("test" <= y) # Checks if 'test' is less than or equal to y (alphabetical order)
```

```
# Comparison of booleans
```

```
print(True > False) # True evaluates to 1, and False evaluates to 0
```

Explanation:

1. In the first comparison, we calculate x as $-3 * 6$, which is -18 . Then we check if $-18 \geq -10$. The result is `False`.
2. In the second comparison, we check if `'test'` is less than or equal to `y`. Since `y` is also `'test'`, the result is `True` as they are equal.
3. The boolean comparison checks if `True` (1) is greater than `False` (0), which is `True`.