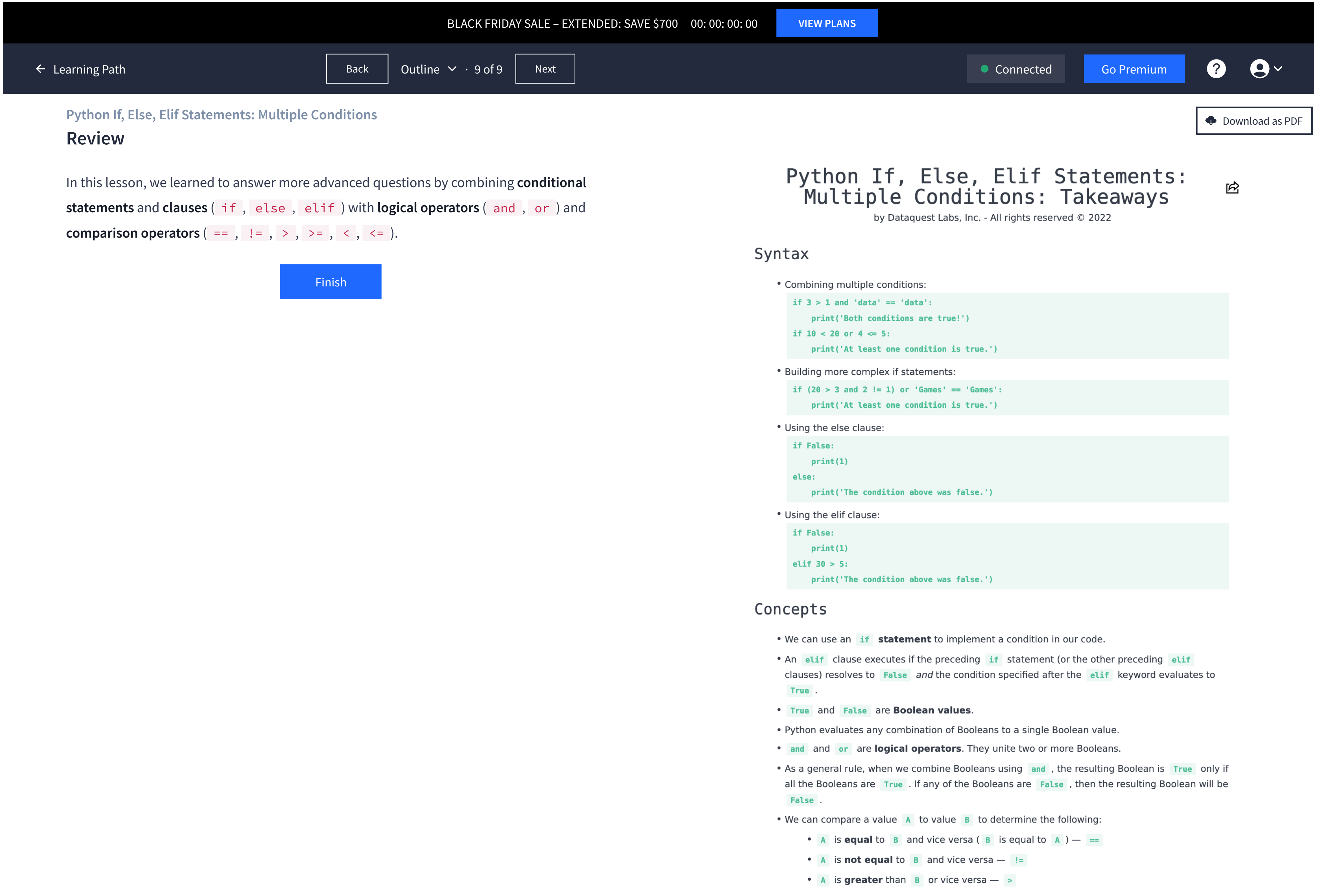

BLACK FRIDAY SALE – EXTENDED: SAVE $700 00: 00: 00: 00 VIEW PLANS

Python If, Else, Elif Statements: Multiple Conditions

Review

In this lesson, we learned to answer more advanced questions by combining **conditional statements** and **clauses** (`if` , `else` , `elif`) with **logical operators** (`and` , `or`) and **comparison operators** (`==` , `!=` , `>` , `>=` , `<` , `<=`).

Python If, Else, Elif Statements: Multiple Conditions: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2022

Syntax

- Combining multiple conditions:

```
if 3 > 1 and 'data' == 'data':
    print('Both conditions are true!')
if 10 < 20 or 4 <= 5:
    print('At least one condition is true.')
```

- Building more complex if statements:

```
if (20 > 3 and 2 != 1) or 'Games' == 'Games':
    print('At least one condition is true.')
```

- Using the else clause:

```
if False:
    print(1)
else:
    print('The condition above was false.')
```

- Using the elif clause:

```
if False:
    print(1)
elif 30 > 5:
    print('The condition above was false.')
```

Concepts

- We can use an `if` **statement** to implement a condition in our code.
- An `elif` clause executes if the preceding `if` statement (or the other preceding `elif` clauses) resolves to `False` *and* the condition specified after the `elif` keyword evaluates to `True` .
- `True` and `False` are **Boolean values**.
- Python evaluates any combination of Booleans to a single Boolean value.
- `and` and `or` are **logical operators**. They unite two or more Booleans.
- As a general rule, when we combine Booleans using `and` , the resulting Boolean is `True` only if all the Booleans are `True` . If any of the Booleans are `False` , then the resulting Boolean will be `False` .
- We can compare a value `A` to value `B` to determine the following:
    - `A` is **equal** to `B` and vice versa (`B` is equal to `A`) — `==`
    - `A` is **not equal** to `B` and vice versa — `!=`
    - `A` is **greater** than `B` or vice versa — `>`