Python try...except...else

Summary: in this tutorial, you'll learn how to use the Python `try...except...else` statement.

Introduction to the Python try...except...else statement

The `try` (https://www.pythontutorial.net/python-basics/python-try-except/) statement has an optional `else` clause with the following syntax:

```
try:
    # code that may cause errors
except:
    # code that handle exceptions
else:
    # code that executes when no exception occurs
```

The `try...except...else` statement works as follows:

- If an exception occurs in the `try` clause, Python skips the rest of the statements in the `try` clause and the `except` statement execute.

- In case no exception occurs in the `try` clause, the `else` clause will execute.

When you include the `finally` (https://www.pythontutorial.net/python-basics/python-try-except-finally/) clause, the `else` clause executes after the `try` clause and before the `finally` clause.

Python try...except...else statement examples

Let's take some examples of using the `try...except...else` statement.

1) Using try...except...else statement for control flow

The following example illustrates how to use the `try...except...else` clause develop a program that calculates the body mass index (BMI).

First, define a function for calculating the (BMI) based on height and weight:

```
def calculate_bmi(height, weight):
    """ calculate body mass index (BMI) """
    return weight / height**2
```

Second, define another function for evaluating BMI:

```
def evaluate_bmi(bmi):
    """ evaluate the bmi """
    if 18.5 <= bmi <= 24.9:
        return 'healthy'

    if bmi >= 25:
        return 'overweight'

    return 'underweight'
```

Third, define a new `main()` function that prompts users for entering height and weight, and prints out the BMI result:

```
def main():
    try:
        height = float(input('Enter your height (meters):'))
        weight = float(input('Enter your weight (kilograms):'))

    except ValueError as error:
        print('Error! please enter a valid number.')
    else:
        bmi = round(calculate_bmi(height, weight), 1)
        evaluation = evaluate_bmi(bmi)

        print(f'Your body mass index is {bmi}')
        print(f'This is considered {evaluation}!')
```

The `main()` function uses the `try...except...else` statement to control its flow. If you enter height and weight that cannot be converted to numbers, the `ValueError` exception will occur.

If no exception occurs, the `else` clause will execute. It calculates the BMI index and displays the evaluation.

Put it all together.

```
def calculate_bmi(height, weight):
    """ calculate body mass index (BMI) """
    return weight / height**2


def evaluate_bmi(bmi):
    """ evaluate the bmi """
    if 18.5 <= bmi <= 24.9:
        return 'healthy'

    if bmi >= 25:
        return 'overweight'
```

```
    return 'underweight'


def main():
    try:
        height = float(input('Enter your height (meters):'))
        weight = float(input('Enter your weight (kilograms):'))

    except ValueError as error:
        print(error)
    else:
        bmi = round(calculate_bmi(height, weight), 1)
        evaluation = evaluate_bmi(bmi)

        print(f'Your body mass index is {bmi}')
        print(f'This is considered {evaluation}!')

main()
```

2) Using Python try...except...else and finally example

The `else` clause executes right before the `finally` (https://www.pythontutorial.net/python-basics/python-try-except-finally/) clause if no exception occurs in the `try` clause.

The following example shows how to use the `try...except...else...finally` clause:

```
fruits = {
    'apple': 10,
    'orange': 20,
    'banana': 30
}

key = None
```

```
while True:
    try:
        key = input('Enter a key to lookup:')
        fruit = fruits[key.lower()]
    except KeyError:
        print(f'Error! {key} does not exist.')
    except KeyboardInterrupt:
        break
    else:
        print(fruit)
    finally:
        print('Press Ctrl-C to exit.')
```

How it works.

- First, define the `fruits` dictionary that contains three elements.
- Second, use a `while` loop to repeatedly get inputs from users. It stops the loop until the users press `Ctrl-C` .
- Third, use the `try...except...else...finally` clause inside the `while` loop. We use the user input to find for the element in the dictionary.

If the key doesn't exist, the `KeyError` exception occurs, the except clause will execute.

If users press `Ctrl-C` , the `KeyboardInterrupt` exception occurs that executes the `break` statement to terminate the loop.

If the key is found in the `fruits` dictionary, the program prints out the found element.

The `finally` clause always executes. It shows the reminder to the users that they should press Ctrl-C to exit.

Summary

- Use the Python `try...except...else` statement provides you with a way to control the flow of the program in case of exceptions.

- The `else` clause executes if no exception occurs in the `try` clause.
- If so, the `else` clause executes after the `try` clause and before the `finally` clause.