Python

How to use

try: except: else: finally:

```python
def division(a, b):
    try:
        # Code that might raise exception
        result = a / b

    except ZeroDivisionError:
        # Handling a specific exception

    except Exception as e:
        # Handling any other exceptions

    else:
        # Runs if try block is successful

    finally:
        # Always executes
```

try:

The try: block in Python contains code that might raise exceptions, allowing for safe execution and handling of potential errors.

try: except:

The except: block is used to catch and handle exceptions raised in the preceding try: block. It specifies what to do when a specific error type occurs.

try: except: else:

The else: block follows a try: except: structure and is executed only if no exceptions are raised in the try: block. It's used for code that should run only if the try block is successful.

try: except: else: finally:

The finally: block always executes after try: and except: blocks, used for resource cleanup or final actions. It runs regardless of whether an exception was raised.

Python

quick tip:

Instead of using a broad ”except Exception:”, pinpoint the exact issue, like ”except ValueError:”. This approach catches errors more effectively and makes your code clearer and more maintainable.

Karl-Fredrik M. Hagman
Happy coding!