Try, except

domingo, 4 de setembro de 2022 20:35

```
try:
    a = 0
    try:
        a = 1/0
    except:
        print('Erro')
except NameError as erro:
    print('Erro do desenvolvedor, fale com ele.')
except (IndexError, KeyError) as erro:
    print('Erro de índice ou chave.')
except Exception as erro:
    print('Ocorreu um erro inesperado.')
else:
    pass
finally:
    a = ''

print(a)
```

Try serve para tentar algo que funcione porém se não der certo existem as except(exceções) podemos definir algum tipo de erro específico ou apenas colocar o exception para tratar todo tipo de erro.

o finally será sempre executado mesmo encontrando erro ou não.

o else vai funcionar se não encontrar nenhum erro.

Quando queremos usar o mesmo erro em outra parte do código usamos da seguinte forma

```
def divide(n1, n2):
    try:
        return n1 / n2
    except ZeroDivisionError as error:
        print('Log: ',error)
        raise

try:
    print(divide(2,0))
except ZeroDivisionError as error:
    print(error)
```

Usando o raise você pode usar novamente em outra parte do codigo o mesmo tipo de erro
Usando o raise também criamos nossa própria exceção

```python
def recebe(n1,n2):
    if n1==0 or n2 ==0:
        raise ValueError("Digite um número que não seja 0")

try:
    mensagem = recebe(2,0)
except ValueError as erro:
    print(erro)
```

```
Digite um número que não seja 0

Process finished with exit code 0
```

Podemos usar try dentro de try, vendo exemplo:

```python
def converte_numero(valor):
    try:
        valor = int(valor)
        return valor
    except ValueError:
        try:
            valor = float(valor)
            return valor
        except ValueError:
            pass


while True:
    numero = converte_numero(input('Digite um número: '))

    if numero is None:
        print('Erro: isso não é um número')
    else:
        print(numero * 2)
```