Professor: Ruan Chaves Machado

Data do Material Didático Criado: 19/05/2025

Data da Aula: 30/05/2025

Tópicos e Temas abordados nessa aula:

* Introdução ao Debugger do VSCode para Python + Uso do Watch
* Como utilizar o Python Debugger: Variáveis em ‘Watch’
* Exercícios de reforço

**Introdução ao Debugger do VSCode para Python + Uso do Watch**

**1. O que é o Debugger do VSCode?**

* O **VSCode Debugger** é uma interface visual que te permite executar seu código passo a passo, inspecionar variáveis, definir breakpoints, avaliar expressões, usar watch, e muito mais, com facilidade, tudo dentro do editor.
* Ele usa internamente o **pdb** para Python, mas oferece muito mais conforto com botões, atalhos, janelas gráficas, etc.
* É uma das ferramentas mais importantes para programadores porque ajuda a entender exatamente o que está acontecendo no código.

**2. Por que usar o Debugger do VSCode?**

* Você não precisa ficar colocando print() no código.
* Pode pausar o código em qualquer linha (breakpoint).
* Explorar o estado do programa (variáveis, pilha de chamadas).
* Modificar variáveis em tempo real.
* Usar **Watch** para monitorar variáveis e expressões ao longo da execução.
* Controlar o fluxo com botões intuitivos (continuar, step over, step into, step out).

## 3. Configurando o Debugger para Python no VSCode

* Instale a extensão oficial do Python da Microsoft no VSCode.
* Abra sua pasta/projeto Python no VSCode.
* Crie um arquivo Python para testar (exemplo: teste.py).
* No menu lateral esquerdo, clique no ícone de **Run and Debug** (▶️ com um inseto).
* Clique em **create a launch.json file** se for solicitado, escolha "Python File".
* Agora o VSCode já está pronto para debugar seu arquivo.

## 4. Como usar o Debugger do VSCode

### Passos básicos:

1. **Setar Breakpoint:** Clique na margem esquerda da linha de código onde quer que o programa pare (aparece um ponto vermelho).
2. **Iniciar Debug:** Clique no botão verde ▶️ no painel Run and Debug ou pressione **F5**.
3. **Controlar execução com botões:**

| **Botão / Atalho** | **Função** |
| --- | --- |
| ▶️ / F5 | Continuar a execução até próximo breakpoint ou fim |
| ⏭️ (Step Over) / F10 | Executa a próxima linha, sem entrar em funções |
| ↪️ (Step Into) / F11 | Entra dentro da função chamada na linha atual |
| ↩️ (Step Out) / Shift+F11 | Sai da função atual, volta para chamador |
| ⏸️ (Pause) | Pausa a execução no ponto atual |
| ⏹️ (Stop) / Shift+F5 | Para o debug |

1. **Inspecionar variáveis:** No painel lateral direito, você vê as variáveis locais e globais atualizadas conforme avança.
2. **Usar Watch:** No painel “WATCH”, você pode adicionar expressões para serem monitoradas continuamente.

## 5. O que é e como usar o Watch no VSCode?

* Watch permite **monitorar variáveis e expressões específicas** enquanto o programa está em execução ou pausado.
* Adicione uma expressão na seção “WATCH” clicando no + e digitando, por exemplo, o nome da variável.
* O valor daquela expressão será atualizado toda vez que o programa pausar.
* Isso ajuda a acompanhar valores importantes sem precisar ficar procurando no painel de variáveis.

**Exemplo 1:**

def fatorial**(**n**):**

resultado **=** 1

**for** i **in** **range(**1**,** n **+** 1**):**

resultado **\*=** i # Coloque breakpoint aqui

**return** resultado

**print(**fatorial**(**5**))**

* Abra no VSCode.
* Defina um breakpoint na linha resultado \*= i.
* Inicie o debug com F5.
* Use os botões Step Over (F10) para ir avançando.
* No painel WATCH, adicione i e resultado.
* Observe como os valores mudam a cada passo.
* Experimente entrar dentro do loop com Step Into (F11), mesmo que aqui não tenha função, para mostrar o funcionamento.

**Exemplo 2: Verificando e Corrigindo Erros em uma Função de Média**

def calcula\_media**(**notas**):**

total **=** 0

**for** nota **in** notas**:**

total **+=** nota

media **=** total **/** **len(**notas**)**

**return** media

notas **=** **[**7**,** 8**,** 6**,** 9**,** 5**]**

**print(**"Média:"**,** calcula\_media**(**notas**))**

### Como usar no Debugger:

* Coloque breakpoint na linha media = total / len(notas).
* Inicie o debug (F5).
* No painel **WATCH**, adicione as variáveis total, notas, media.
* Use **Step Over (F10)** para avançar linha a linha, acompanhando os valores.
* Explore o que acontece quando notas estiver vazia (adicione teste notas = [] para ver erro).