

Trường: Agentra

Tổ: Công nghệ thông tin

Họ và tên giáo viên: Phúc Hảo

CẤU TRÚC RỄ NHÁNH TRONG PYTHON

Môn: Tin học Lớp: 10

Thời gian thực hiện: (1 tiết)

I. MỤC TIÊU

1. Về kiến thức

Sau bài học, học sinh:

- Nêu được khái niệm cấu trúc rẽ nhánh và vai trò của nó trong lập trình.
- Trình bày được cú pháp của lệnh `if`, `if...else` và `if...elif...else` trong Python.
- Giải thích được nguyên lý hoạt động của cấu trúc rẽ nhánh: kiểm tra điều kiện Boolean và thực hiện khối lệnh tương ứng.
- Xác định được tầm quan trọng của thụt lề (indentation) trong Python khi viết cấu trúc rẽ nhánh.

2. Về năng lực

a) Năng lực chung:

- **Tự chủ và tự học:** Chủ động đọc tài liệu, nghiên cứu ví dụ mẫu để hiểu cú pháp cấu trúc rẽ nhánh; tự kiểm tra lỗi và sửa lỗi khi chạy chương trình.
- **Giao tiếp và hợp tác:** Thảo luận theo cặp hoặc nhóm nhỏ để phân tích bài toán và lựa chọn cấu trúc rẽ nhánh phù hợp; trình bày, bảo vệ ý kiến trước lớp một cách rõ ràng.
- **Giải quyết vấn đề và sáng tạo:** Phân tích bài toán thực tế, xác định điều kiện, thiết kế thuật toán rẽ nhánh và viết mã Python để giải quyết vấn đề đặt ra.

b) Năng lực đặc thù (Tin học):

- **Tư duy lập trình (Computational Thinking):** Phân tích bài toán thành các trường hợp phân biệt, diễn đạt điều kiện rẽ nhánh dưới dạng biểu thức so sánh và logic.

- **Giải quyết vấn đề với sự hỗ trợ của công nghệ:** Sử dụng môi trường Python để lập trình cấu trúc rẽ nhánh, chạy thử với nhiều bộ dữ liệu và sửa lỗi chương trình.
- **Ứng dụng tin học:** Nhận biết ứng dụng thực tế của cấu trúc rẽ nhánh trong các bài toán phân loại, kiểm tra điều kiện và ra quyết định tự động.

c) *Năng lực số phát triển qua bài:*

- **Sử dụng công cụ số để giải quyết vấn đề:** Học sinh sử dụng Python IDLE hoặc môi trường lập trình trực tuyến Replit để viết, chạy và kiểm tra chương trình có cấu trúc rẽ nhánh; thử nghiệm với nhiều giá trị đầu vào khác nhau để xác minh tính đúng đắn của từng nhánh lệnh.
- **Khai thác thông tin số:** Tìm kiếm và tham khảo tài liệu Python chính thức (docs.python.org) để tra cứu cú pháp khi gặp lỗi hoặc cần mở rộng kiến thức.

3. Về phẩm chất

- **Trung thực:** Tự giác thực hành, không sao chép bài của bạn; kiểm tra kết quả chương trình một cách khách quan và báo cáo đúng kết quả.
- **Chăm chỉ:** Tích cực tham gia các hoạt động học tập, không ngại thử nghiệm, sửa lỗi và hoàn thiện chương trình cho đến khi chạy đúng.
- **Trách nhiệm:** Hoàn thành nhiệm vụ được giao trong thời gian quy định; chia sẻ kết quả và hỗ trợ bạn trong nhóm một cách nghiêm túc.

II. THIẾT BỊ DẠY HỌC VÀ HỌC LIỆU

1. Giáo viên:

- Máy tính có cài Python 3 và phần mềm trình chiếu (PowerPoint hoặc Google Slides).
- Máy chiếu (projector), bảng trắng, bút dạ.
- Slide bài giảng thể hiện sơ đồ khối (flowchart) của ba dạng cấu trúc rẽ nhánh.
- Phiếu học tập (PHT) in sẵn đề bài luyện tập và ô chờ điền code.

2. Học sinh:

- Sách giáo khoa Tin học 10 (Bộ Kết nối tri thức với cuộc sống hoặc Cánh diều).

- Vở ghi, bút; máy tính (laptop hoặc máy tính để bàn tại phòng máy).
- Tài khoản Replit đã đăng nhập sẵn (nếu có kết nối Internet) hoặc Python IDLE đã cài đặt.

3. Học liệu số:

- Tài liệu Python chính thức: docs.python.org/3/tutorial/controlflow.html.
- Video minh họa ngắn (2 phút) về luồng thực thi của `if...elif...else` — GV chuẩn bị sẵn đường dẫn hoặc trình chiếu trực tiếp trong tiết học.

III. TIỀN TRÌNH DẠY HỌC

Hoạt động 1: Mở đầu — Khởi động (5 phút)

a) Mục tiêu:

- Tạo tình huống có vấn đề, kích thích học sinh nhận ra sự cần thiết của việc ra quyết định (rẽ nhánh) trong lập trình.
- Kết nối kinh nghiệm thực tế của học sinh với khái niệm cấu trúc rẽ nhánh sẽ học trong bài.

b) Nội dung:

GV trình chiếu tình huống: *“Một chương trình cần kiểm tra số nguyên n nhập từ bàn phím. Nếu n chia hết cho 2, in ra "Số chẵn"; ngược lại, in ra "Số lẻ". Bạn sẽ viết thế nào?”*

GV đặt hai câu hỏi gợi mở:

- Trong cuộc sống, bạn thường ra quyết định như thế nào khi đứng trước hai hoặc nhiều lựa chọn?
- Máy tính có thể “tự quyết định” như người không? Nó cần thông tin gì để lựa chọn hành động?

c) Sản phẩm: Câu trả lời ban đầu của học sinh về tình huống: ý kiến cá nhân về cách chương trình xử lý điều kiện. Học sinh bắt đầu nhận ra cần có một cấu trúc đặc biệt để ra quyết định trong lập trình.

d) Tổ chức thực hiện:

Hoạt động của GV và HS	Sản phẩm
Bước 1: Chuyển giao nhiệm vụ - GV chiếu tình huống bài toán kiểm tra số chẵn/lẻ lên màn hình. - GV đặt 2 câu hỏi gợi mở về cách ra quyết định trong cuộc sống và trong máy tính. - GV yêu cầu HS suy nghĩ cá nhân trong 1 phút, sau đó phát biểu nhanh.	- HS đọc tình huống và nhận câu hỏi từ GV. - HS chuẩn bị ý kiến cá nhân trước khi phát biểu.
Bước 2: Thực hiện nhiệm vụ - HS suy nghĩ độc lập, không trao đổi trong 1 phút. - GV quan sát, không can thiệp; tạo không khí thoải mái để HS tự do bày tỏ suy nghĩ ban đầu.	- HS hình thành suy nghĩ ban đầu: máy tính cần kiểm tra một điều kiện nào đó rồi chọn hành động tương ứng (nếu đúng thì làm A, nếu sai thì làm B).
Bước 3: Báo cáo, thảo luận - GV mời 2-3 HS phát biểu ý kiến. - GV ghi tóm tắt ý kiến HS lên bảng (không nhận xét đúng/sai ngay).	- HS trình bày: máy tính cần so sánh, kiểm tra điều kiện; nếu điều kiện đúng thì thực hiện hành động A, nếu sai thì thực hiện hành động B.
Bước 4: Kết luận, nhận định - GV chốt: để điều khiển luồng thực thi theo điều kiện, lập trình dùng cấu trúc rẽ nhánh . - GV dẫn dắt vào nội dung bài: Python cung cấp lệnh <code>if</code> để thực hiện điều này.	Kết luận khởi động: Cấu trúc rẽ nhánh là công cụ lập trình cho phép chương trình lựa chọn và thực hiện các hành động khác nhau tùy theo điều kiện đúng hay sai. Python dùng từ khoá <code>if</code> để xây dựng cấu trúc này.

Hoạt động 2: Hình thành kiến thức mới (20 phút)

a) Mục tiêu:

- Nêu được cú pháp và ý nghĩa của ba dạng cấu trúc rẽ nhánh: `if`, `if...else`, `if...elif...else`.
- Giải thích được vai trò bắt buộc của thụt lề (indentation) trong Python.
- Đọc hiểu và viết được đoạn mã Python đơn giản có cấu trúc rẽ nhánh, chạy thử trên máy tính.

b) Nội dung:

Kiến thức 1: Cấu trúc `if` đơn

Cú pháp:

```
if <điều_kiện>:  
    <khối_lệnh>
```

Trong đó <điều_kiện> là biểu thức có giá trị `True` hoặc `False`. Khối lệnh chỉ được thực thi khi điều kiện là `True`. Nếu điều kiện là `False`, Python bỏ qua toàn bộ khối lệnh đó và tiếp tục chương trình.

Kiến thức 2: Cấu trúc `if...else`

Cú pháp:

```
if <điều_kiện>:  
    <khối_lệnh_1>  
else:  
    <khối_lệnh_2>
```

Nếu điều kiện là `True`, thực hiện khối lệnh 1; ngược lại (= `False`), thực hiện khối lệnh 2. Đảm bảo đúng một trong hai nhánh luôn được thực hiện.

Kiến thức 3: Cấu trúc `if...elif...else`

Cú pháp:

```
if <điều_kiện_1>:  
    <khối_lệnh_1>  
elif <điều_kiện_2>:  
    <khối_lệnh_2>  
else:  
    <khối_lệnh_3>
```

Python lần lượt kiểm tra các điều kiện từ trên xuống. Khi gặp điều kiện đúng, thực hiện khối lệnh tương ứng và bỏ qua tất cả các nhánh còn lại. Nếu không có điều kiện nào đúng, thực hiện khối **else**. Có thể có nhiều **elif** liên tiếp.

Lưu ý quan trọng về thụt lề: Python sử dụng thụt lề — thường là 4 dấu cách hoặc 1 phím Tab — để xác định khối lệnh thuộc về nhánh nào. Thiếu hoặc sai thụt lề sẽ gây lỗi **IndentationError** và chương trình không chạy được.

c) Sản phẩm: Học sinh ghi được vào vở: cú pháp ba dạng cấu trúc rẽ nhánh, quy tắc thụt lề trong Python, ít nhất một ví dụ minh họa cho mỗi dạng. Học sinh chạy thử được ít nhất một chương trình có cấu trúc rẽ nhánh trên máy tính.

d) Tổ chức thực hiện:

Hoạt động của GV và HS	Sản phẩm
Bước 1: Chuyển giao nhiệm vụ - GV trình chiếu lần lượt cú pháp if, if...else, if...elif...else kèm sơ đồ khối minh họa. - GV chiếu ví dụ code kiểm tra số chẵn/lẻ (if...else) và ví dụ phân loại điểm học lực (if...elif...else). - GV yêu cầu HS: đọc code mẫu, dự đoán kết quả in ra khi nhập các giá trị khác nhau, sau đó chạy thử trên Replit/Python IDLE để kiểm chứng.	- HS nhận phiếu học tập có in sẵn code mẫu và ô câu hỏi phân tích. - HS mở Replit hoặc Python IDLE, sẵn sàng nhập và chạy code (NLS: sử dụng công cụ số).
Bước 2: Thực hiện nhiệm vụ - HS đọc từng đoạn code mẫu, theo dõi luồng thực thi theo sơ đồ khối GV chiếu. - HS nhập code lên máy tính, chạy thử với nhiều giá trị đầu vào, ghi lại kết quả vào PHT. - GV cố tình viết một ví dụ bị sai thụt lề, cho HS thấy thông báo IndentationError và lý giải nguyên nhân.	- HS chạy được chương trình kiểm tra số chẵn/lẻ và chương trình phân loại điểm. - HS ghi nhận: thiếu thụt lề gây lỗi; Python không dùng dấu ngoặc nhọn như C/C++ mà dùng thụt lề để xác định khối lệnh.

Bước 3: Báo cáo, thảo luận - GV mời 1–2 HS lên bảng viết lại cú pháp từng dạng và giải thích ý nghĩa từng phần. - GV đặt câu hỏi: “Nếu bỏ hoàn toàn phần <code>else</code>, điều gì xảy ra khi điều kiện sai?” - Cả lớp thảo luận, bổ sung.	- HS trình bày đúng cú pháp ba dạng cấu trúc trên bảng. - HS giải thích: bỏ <code>else</code> thì khi tất cả điều kiện đều sai, không có hành động nào được thực thi; chương trình tiếp tục bình thường.
Bước 4: Kết luận, nhận định - GV chốt kiến thức trọng tâm, nhấn mạnh quy tắc thực lệ bắt buộc. - GV lưu ý: trong <code>if...elif...else</code>, thứ tự các điều kiện ảnh hưởng đến kết quả; cần sắp xếp hợp lý.	Kiến thức trọng tâm: (1) <code>if <đk></code>: — thực hiện khối lệnh khi <code><đk></code> là <code>True</code>. (2) <code>if...else</code> — chọn một trong hai nhánh. (3) <code>if...elif...else</code> — chọn một trong nhiều nhánh theo thứ tự kiểm tra từ trên xuống. (4) Thực lệ 4 dấu cách là bắt buộc trong Python.

Hoạt động 3: Luyện tập (15 phút)

a) Mục tiêu:

- Vận dụng được cú pháp `if`, `if...else`, `if...elif...else` để lập trình giải các bài toán đơn giản.
- Nhận ra và sửa được lỗi cú pháp thường gặp: quên dấu `:`, sai thực lệ, sai thứ tự điều kiện trong `elif`.

b) Nội dung:

HS thực hành hai bài tập trên máy tính (hoặc viết tay nếu không có phòng máy):

Bài 1 (5 phút): Viết chương trình nhập vào một số nguyên n . In ra "Dương" nếu $n > 0$; in ra "Âm" nếu $n < 0$; in ra "Bằng không" nếu $n = 0$.

Bài 2 (7 phút): Viết chương trình nhập vào điểm trung bình d (với $0 \leq d \leq 10$). Phân loại học lực theo quy tắc:

- $d \geq 8,0$: in ra "Giỏi"
- $6,5 \leq d < 8,0$: in ra "Khá"
- $5,0 \leq d < 6,5$: in ra "Trung bình"
- $d < 5,0$: in ra "Yếu"

Sau khi viết xong, HS kiểm tra chương trình với ít nhất 3 bộ dữ liệu thử nghiệm khác nhau cho mỗi bài.

c) Sản phẩm: File .py (hoặc code viết trên phiếu học tập) của hai bài tập; kết quả chạy đúng với ít nhất 3 bộ dữ liệu thử nghiệm cho mỗi bài (ghi kết quả vào cột kiểm thử trong PHT).

d) Tổ chức thực hiện:

Hoạt động của GV và HS	Sản phẩm
Bước 1: Chuyển giao nhiệm vụ - GV phát phiếu học tập ghi rõ đề bài 1 và bài 2, có cột ghi kết quả kiểm thử. - GV hướng dẫn thứ tự: làm bài 1 trước (5 phút), rồi bài 2 (7 phút), kiểm thử cuối (3 phút). - GV nhắc: trước khi code, hãy viết ra giấy các điều kiện rẽ nhánh cần dùng.	- HS nhận đề bài, xác định dạng cấu trúc cần dùng (<code>if...elif...else</code> cho cả hai bài). - HS phân tích điều kiện trên giấy trước khi gõ code.
Bước 2: Thực hiện nhiệm vụ - HS làm việc cá nhân, viết code trên máy tính hoặc trên phiếu học tập. - GV đi quan sát từng bàn, chú ý các lỗi phổ biến: quên dấu : sau điều kiện, sai thụt lề, đặt điều kiện $d \geq 6.5$ trước $d \geq 8.0$ dẫn đến nhảy nhầm nhánh. - GV gợi ý (không cho đáp án): “Em kiểm tra thụt lề chưa?”, “Điều kiện nào cần được kiểm tra trước?”	- HS viết và chạy được bài 1 với 3 giá trị thử: $n = 5$, $n = -3$, $n = 0$. - HS hoàn thiện bài 2, kiểm thử với $d = 9,0$; $d = 7,0$; $d = 5,5$; $d = 3,5$.

Bước 3: Báo cáo, thảo luận - GV mời 1 HS chiếu (hoặc chép lên bảng) code bài 1, 1 HS khác trình bày code bài 2. - Cả lớp nhận xét: code đúng chưa? Thứ tự điều kiện có hợp lý không? - GV chỉ ra lỗi điển hình (nếu có) và giải thích nguyên nhân.	- HS nhận xét: nếu đặt <code>elif d >= 6.5</code> trước <code>elif d >= 8.0</code> thì điểm 9 cũng bị xếp “Khá” — thứ tự kiểm tra <code>elif</code> ảnh hưởng trực tiếp đến kết quả. - HS rút ra bài học: trong <code>if...elif...else</code>, luôn kiểm tra điều kiện chặt hơn trước.
Bước 4: Kết luận, nhận định - GV trình chiếu đáp án chuẩn cả hai bài. - GV chốt: khi dùng <code>elif</code>, cần sắp xếp điều kiện theo thứ tự từ ngưỡng cao xuống ngưỡng thấp (hoặc ngược lại tùy cách viết) để tránh rơi nhầm nhánh.	Đáp án bài 1: <code>if n > 0</code> → in “Dương”; <code>elif n < 0</code> → in “Âm”; <code>else</code> → in “Bằng không”. Đáp án bài 2: kiểm tra theo thứ tự <code>if d >= 8.0</code> → Giỏi; <code>elif d >= 6.5</code> → Khá; <code>elif d >= 5.0</code> → Trung bình; <code>else</code> → Yếu.

Hoạt động 4: Vận dụng (5 phút)

a) Mục tiêu:

- Kết nối kiến thức cấu trúc rẽ nhánh với bài toán thực tiễn gần gũi trong cuộc sống.
- Khơi gợi hứng thú tự học, định hướng học sinh tự khám phá và hoàn thiện ở nhà.

b) Nội dung:

GV trình chiếu bài toán vận dụng:

Một cửa hàng áp dụng chính sách giảm giá như sau: hoá đơn từ 500.000 đồng trở lên được giảm 10%; hoá đơn từ 200.000 đồng đến dưới 500.000 đồng được giảm 5%; hoá đơn dưới 200.000 đồng không được giảm. Viết chương trình Python nhập số tiền hoá đơn và in ra số tiền thực tế phải trả.

HS làm việc theo cặp (2 phút), nêu ý tưởng (không cần viết code hoàn chỉnh trong lớp):

- Xác định ba điều kiện rẽ nhánh cần dùng.

- Xác định công thức tính số tiền phải trả: nếu giảm $r\%$ thì tiền trả = hóa đơn $\times (1 - r/100)$.

GV giao bài tập về nhà: hoàn thiện code Python cho bài toán của hàng trên; kiểm thử với hóa đơn 100.000 đồng, 300.000 đồng và 600.000 đồng; nộp file .py vào đầu tiết học sau.

c) Sản phẩm: Học sinh nêu được ý tưởng điều kiện rẽ nhánh và công thức tính cho bài toán của hàng; nhận bài tập về nhà cụ thể để hoàn thiện và nộp bài.

d) Tổ chức thực hiện:

Hoạt động của GV và HS	Sản phẩm
Bước 1: Chuyển giao nhiệm vụ - GV trình chiếu bài toán của hàng giảm giá lên màn hình. - GV yêu cầu HS làm việc theo cặp (2 phút): xác định ba điều kiện rẽ nhánh và viết sơ đồ ý tưởng (không cần code hoàn chỉnh). - GV gợi ý: ghi ra giấy “Nếu ... thì giảm ...” cho từng trường hợp.	- HS đọc đề bài, ghép cặp và bắt đầu phân tích bài toán. - HS xác định cần dùng cấu trúc <code>if...elif...else</code> với 3 nhánh.
Bước 2: Thực hiện nhiệm vụ - HS thảo luận cặp đôi: xác định ba điều kiện, tỉ lệ giảm tương ứng và công thức tính tiền mỗi trường hợp. - GV quan sát, gợi ý nếu HS lúng túng ở công thức: giảm 10% $\Rightarrow$ trả 90% $\Rightarrow$ nhân 0,9; giảm 5% $\Rightarrow$ trả 95% $\Rightarrow$ nhân 0,95.	- HS xác định được: <code>if hoa_don >= 500000</code> (giảm 10%); <code>elif hoa_don >= 200000</code> (giảm 5%); <code>else</code> (không giảm). - HS ghi được công thức tính tiền cho từng nhánh.
Bước 3: Báo cáo, thảo luận - GV mời 1 cặp HS trình bày ý tưởng: điều kiện từng nhánh và công thức tương ứng. - GV cho cả lớp nhận xét nhanh: điều kiện có đủ và đúng thứ tự không?	- HS trình bày: ba điều kiện, ba công thức tính tiền phải trả. - Cả lớp xác nhận thứ tự kiểm tra điều kiện từ ngưỡng cao (≥ 500.000) xuống thấp hơn (≥ 200.000) là đúng.

Bước 4: Kết luận, nhận định - GV chốt ý tưởng đúng, nhấn mạnh: bài toán thực tế cũng chỉ cần <code>if...elif...else</code> là giải được. - GV giao bài tập về nhà cụ thể và nhắc lịch nộp bài. - GV dặn dò: tiết sau học cấu trúc lặp trong Python (<code>for</code>, <code>while</code>).	Bài tập về nhà: Hoàn thiện code Python cho bài toán của hàng giảm giá; kiểm thử với ba giá trị: 100.000 đồng (không giảm), 300.000 đồng (giảm 5%), 600.000 đồng (giảm 10%); ghi kết quả kiểm thử vào file và nộp <code>.py</code> vào đầu tiết học sau.
---	---

....., ngàythángnăm

TỔ TRƯỞNG CHUYÊN MÔN

(Ký và ghi rõ họ tên)

GIÁO VIÊN

(Ký và ghi rõ họ tên)

Phúc Hảo