

Topic 1: Complex Conditional Patterns

أنماط الشروط المتقدمة في جافاسكريبت

Variable Declarations

[إعلان المتغيرات]

```
let number = 10;      // قيمة قابلة للتغيير
const PI = 3.14159;   // قيمة ثابتة لا يمكن تغييرها
```

var للثوابت التي لا تتغير، و const، للمتغيرات التي قد تتغير قيمتها: let في جافاسكريبت، ثلاث طرق لإعلان المتغيرات وهي الطريقة القديمة التي لا ينصح باستخدامها في الكود الحديث

```
%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart LR
    A[Type] --> B[name] --> C[=] --> D[value]
    style A fill:#f9f,stroke:#333,stroke-width:1px
    style D fill:#bbf,stroke:#333,stroke-width:1px
```

If-Else If-Else Statement

[جملة if-else if-else]

```
let score = 85;

if (score >= 90) {
    console.log("ممتاز");
} else if (score >= 80) {
    console.log("جيد جداً");
} else if (score >= 70) {
    console.log("جيد");
} else {
    console.log("بحاجة إلى تحسين");
}
```

عندما نريد فحص عدة شروط بالترتيب. يتم تنفيذ الكود المرتبط بأول شرط صحيح فقط if-else if-else تستخدم جملة

```
%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart TB
    A[if] --> B[condition 1]
    B -->|true| C[action 1]
    B -->|false| D[else if]
    D --> E[condition 2]
    E -->|true| F[action 2]
```

```

E -->|false| G[else]
G --> H[default action]

style A fill:#f9f,stroke:#333,stroke-width:1px
style B fill:#bbf,stroke:#333,stroke-width:1px
style D fill:#f9f,stroke:#333,stroke-width:1px
style E fill:#bbf,stroke:#333,stroke-width:1px

```

Switch Statement

[جملة التبديل]

```

let day = "Tuesday";

switch (day) {
  case "Monday":
    console.log("بداية الأسبوع");
    break;
  case "Friday":
    console.log("نهاية الأسبوع");
    break;
  default:
    console.log("أيام الأسبوع العادية");
}

```

if/else تستخدم عندما نريد مقارنة قيمة واحدة بعدة قيم مختلفة. أكثر تنظيماً من استخدام عدة switch جملة

```

%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart TB
  A[switch] --> B[variable]
  B --> C{case 1}
  C -->|match| D[action 1]
  D --> E[break]
  C -->|no match| F{case 2}
  F -->|match| G[action 2]
  G --> H[break]
  F -->|no match| I[default]

style A fill:#f9f,stroke:#333,stroke-width:1px
style C fill:#bbf,stroke:#333,stroke-width:1px
style E fill:#f96,stroke:#333,stroke-width:1px
style F fill:#bbf,stroke:#333,stroke-width:1px
style H fill:#f96,stroke:#333,stroke-width:1px

```

Topic 2: Loops in JavaScript

الحلقات التكرارية في جافاسكريبت

While Loop

[while loop]

```
let counter = 10;

while (counter == 10) {
  console.log(counter);
  counter = counter - 1;
}
```

تستمر في التنفيذ طالما الشرط صحيح. يجب تغيير قيمة الشرط داخل الحلقة لتجنب الحلقات اللانهائية while حلقة

```
%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart TB
    A[Init var] --> B[while]
    B --> C{condition}
    C -->|true| D[code block]
    D --> E[update var]
    E --> C
    C -->|false| F[end]

    style B fill:#f9f,stroke:#333,stroke-width:1px
    style C fill:#bbf,stroke:#333,stroke-width:1px
    style E fill:#f96,stroke:#333,stroke-width:1px
```

Countdown With While Loop

[while العد التنازلي باستخدام حلقة]

```
let num = 100;
while (num >= 0) {
  console.log(num);
  num = num - 2;
}
```

للعَد التنازلي، مثل هذا المثال الذي يعد تنازلياً من 100 إلى 0 بخطوات من 2 while يمكننا استخدام حلقة

```
%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart TB
    A[num = 100] --> B[while]
    B --> C{num >= 0}
    C -->|true| D[log num]
    D --> E[num -= 2]
    E --> C
    C -->|false| F[end]
```

```
style B fill:#f9f,stroke:#333,stroke-width:1px
style C fill:#bbf,stroke:#333,stroke-width:1px
style E fill:#f96,stroke:#333,stroke-width:1px
```

Loop Control

[loops التحكم في ال]

```
let i = 1;
while (i <= 10) {
  if (i === 5) {
    i++; // 5 تخطي الرقم
    continue;
  }
  if (i === 8) {
    break; // 8 الخروج من الحلقة عند الوصول للرقم
  }
  console.log(i);
  i++;
}
```

لتخطي الدورة الحالية والانتقال للدورة التالية continue للخروج من الحلقة، و break يمكننا استخدام

```
%%{init: {'theme': 'base', 'themeVariables': { 'fontSize': '10px'}}}%
flowchart TB
    A[i = 1] --> B[while]
    B --> C{i <= 10}
    C -->|true| D{i === 5}
    D -->|true| E[i++]
    E --> F[continue]
    F --> C
    D -->|false| G{i === 8}
    G -->|true| H[break]
    G -->|false| I[log i]
    I --> J[i++]
    J --> C
    C -->|false| K[end]

    style B fill:#f9f,stroke:#333,stroke-width:1px
    style C fill:#bbf,stroke:#333,stroke-width:1px
    style F fill:#f96,stroke:#333,stroke-width:1px
    style H fill:#f96,stroke:#333,stroke-width:1px
```

Practical Exercise

تمرين عملي: طباعة جدول الضرب

```
// طباعة جدول ضرب الرقم 5
let number = 5;

for (let i = 1; i <= 10; i++) {
  console.log(number + " x " + i + " = " + (number * i));
}
```

Key Concepts Covered

المفاهيم الرئيسية التي تم تغطيتها

1. **var و const و let إعلان المتغيرات:** استخدام
2. **if-else if-else جملة:** فحص عدة شروط بالترتيب
3. **switch جملة:** طريقة منظمة للتحقق من عدة قيم
4. **while:** تنفيذ كود بشكل متكرر طالما الشرط صحيح
5. **break و continue الحلقات:** التحكم في سير الحلقات