

Bài thực hành 3 - Javascript



Chèn javascript vào tài liệu HTML

- Viết giữa cặp thẻ `<script></script>` ở bất cứ nơi đâu trong tài liệu HTML.
- Gắn tập tin `*.js` vào tài liệu HTML

```
<script src="tập-tin-js"></script>
```



Cấu trúc điều khiển

- `if/else`
- `switch/case`
- `while`
- `do/while`
- `for`
- `for..in`



Number

Kiểm tra giá trị là số.

- `isNaN()`: kiểm tra một giá trị có phải số hay không.
- `isInteger()`: kiểm tra một giá trị có phải số nguyên hay không.

Chuyển số thành chuỗi.

- `toString()`: chuyển số thành chuỗi.
- `toFixed()`: chuyển số thành chuỗi với số chữ số thập phân được chỉ định.
- `toPrecision()`: chuyển số thành chuỗi với chiều dài chỉ định.

Chuyển chuỗi thành số.

- `Number()`: chuyển chuỗi số thành số.
- `parseFloat()`: chuyển chuỗi số thành số thực.
- `parseInt()`: chuyển chuỗi số thành số nguyên.



Math

- `Math.abs()`: trị tuyệt đối.
- `Math.sqrt()`: căn bậc 2.
- `Math.pow(a, b)`: a^b .
- `Math.max()`: số lớn nhất các số.
- `Math.min()`: số nhỏ nhất các số.
- `Math.random()`: sinh số ngẫu nhiên từ 0 đến 1.
- `Math.ceil(a)`: số nguyên nhỏ nhất + lớn hơn hoặc bằng số a.
- `Math.floor(a)`: số nguyên lớn nhất + nhỏ hơn hoặc bằng số a.
- `Math.round(a)`: số nguyên gần với số a nhất.



Mảng

- Trong Javascript, mảng có thể chứa các phần tử có kiểu dữ liệu khác nhau.
- Tạo mảng theo 1 trong 2 cách

```
var a = [1, 2, 5, 6]; // thường dùng
```

```
var b = new Array(1, 2, 5, 6); // hạn chế dùng
```

- Chỉ số mảng tính từ 0, để truy xuất phần tử thứ i trong mảng a dùng cú pháp: $a[i]$
- Lấy chiều dài mảng a : $a.length$

Vài phương thức phổ biến: cho mảng a

- $a.push()$: thêm một hoặc nhiều phần tử cuối mảng a .
- $a.pop()$: lấy phần tử ở cuối ra khỏi mảng a .
- $a.unshift()$: thêm một hoặc nhiều phần tử đầu mảng a .
- $a.shift()$: lấy phần tử ở đầu ra khỏi mảng a .
- $a.indexOf(x, fromIndex)$: tìm vị trí đầu tiên xuất hiện x trong mảng a tính từ vị trí $fromIndex$ (mặc định là 0), và trả về -1 nếu không tìm thấy.
- $a.lastIndexOf(x, endIndex)$: trả về vị trí cuối cùng tìm thấy x tính từ đầu mảng đến $endIndex - 1$ ($endIndex$ mặc định là chiều dài mảng).
- $a.reverse()$: đảo ngược mảng a .
- $a.sort()$: sắp xếp mảng a .
- $a.slice(beginIndex, endIndex)$: cắt 1 đoạn trong mảng từ $beginIndex$ đến $endIndex - 1$. Nếu các chỉ số truyền vào là âm sẽ được tính từ cuối mảng.
- $a.concat()$: nối các phần tử hoặc mảng khác vào mảng a .
- $a.join(seperator)$: chuyển mảng thành một chuỗi với $seperator$ là chuỗi phân tách giữa các phần tử.
- $a.splice(fromIndex, removingNum, addingNumbers)$: từ vị trí $fromIndex$ xóa $removingNum$ phần tử và thêm các phần tử $addingNumbers$ vào mảng.
- $a.forEach(function(value, index) {})$: lặp qua các phần tử mảng.



Chuỗi

- Chuỗi trong Javascript được đặt trong cặp dấu nháy đơn (') hoặc nháy kép (").
- Lấy chiều dài chuỗi s : $s.length$

Các phương thức thông dụng: cho chuỗi s

- $s.charAt(idx)$: trả về ký tự vị trí idx .
- $s.charCodeAt(idx)$: trả về mã ASCII của ký tự vị trí idx .
- $s.toLowerCase()$: chuyển chuỗi thành các ký tự thường.
- $s.toUpperCase()$: chuyển chuỗi thành các ký tự hoa.
- $s.concat(s1)$: nối chuỗi s và $s1$.
- $s.indexOf(s1, fromIndex)$: tìm vị trí đầu tiên xuất hiện chuỗi $s1$ trong chuỗi s tính từ vị trí $fromIndex$ (mặc định $fromIndex = 0$).
- $s.lastIndexOf(s1, endIndex)$: tìm vị trí cuối cùng xuất hiện chuỗi $s1$ trong chuỗi s tính từ đầu đến $endIndex - 1$ (mặc định $endIndex$ là chiều dài chuỗi).
- $s.substring(fromIndex, endIndex)$: trả về chuỗi con từ $fromIndex$ đến $endIndex - 1$ ($fromIndex$ và $endIndex$ không thể nhận giá trị âm).
- $s.substr(fromIndex, len)$: trả về chuỗi con có chiều dài là len tính từ $fromIndex$ ($fromIndex$ có thể âm, len mặc định là chiều dài chuỗi).
- $s.slice(fromIndex, endIndex)$: trả về chuỗi con từ $fromIndex$ đến $endIndex - 1$ ($fromIndex$ và $endIndex$ có thể âm).

- o `s.split(seperator)`: tách chuỗi thành mảng các chuỗi với dấu hiệu tách là `seperator`.
- o `s.replace(oldStr, newStr)`: thay thế chuỗi `oldStr` thành chuỗi `newStr`.
- o `s.search(re)`: tìm chuỗi con khớp với một biểu thức chính quy.



HTML DOM

Tìm thành phần HTML

- o `document.getElementById(id)`: tìm thành phần dựa trên id và trả về `undefined` nếu không tìm thấy.
- o `document.getElementsByTagName(tag)`: tìm dựa trên tên thành phần.
- o `document.getElementsByClassName(class)`: tìm dựa trên giá trị của thuộc tính `class`.
- o `document.getElementsByName(name)`: tìm dựa trên giá trị thuộc tính `name`.
- o `document.querySelectorAll(selector)`: tìm dựa trên cú pháp `selector` giống CSS.

Thay đổi nội dung thành phần HTML: e là một thành phần HTML

- o Thay đổi nội dung HTML.
`e.innerHTML = <new-html>;`
- o Thay đổi giá trị thuộc tính.
`e.<attribute> = <value>;`
`e.setAttribute(<attribute>, <value>);`
- o Thay đổi css
`e.style.<property> = <new-style>;`
- o Thêm xử lý sự kiện:
`e.<event> = function() {};`
- o Lắng nghe sự kiện:
`e.addEventListener(event, function, useCapture);`

Tạo thành phần HTML

- o `document.createElement(e)`
- o `document.createTextNode(e)`

Thêm thành phần HTML: `document.appendChild(e)`

Xóa thành phần HTML: `document.removeChild(e)`



Tạo hiệu ứng

- o `setTimeout(function, duration)`: function sẽ được gọi sau khoảng thời gian `duration`.
- o `setInterval(function, duration)`: function sẽ được gọi sau mỗi khoảng thời gian `duration`.
- o `clearTimeout(timeout-variable)`: xóa timeout đã thiết lập.
- o `clearInterval(interval-variable)`: xóa interval đã thiết lập.



Dialog:

- o `window.alert()`: hiển thị một thông báo.

- `window.confirm()` : cần sự xác nhận của người dùng.
- `window.prompt()` : tương tác với người dùng.

Bài 1: Thiết kế giao diện HTML và viết xử lý javascript giải các bài toán sau:

a) Giải và biện luận phương trình bậc hai: $ax^2 + bx + c = 0$

Biết: $\Delta = b^2 - 4ac$

Nếu $\Delta < 0$: Phương trình vô nghiệm

Nếu $\Delta = 0$: Phương trình có nghiệm kép

$$x_1 = x_2 = \frac{-b}{2a}$$

Nếu $\Delta > 0$: Phương trình có hai nghiệm phân biệt

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a} \text{ và } x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

Giải phương trình $ax^2 + bx + c = 0$

Hệ số a

Hệ số b

Hệ số c

Giải phương trình $ax^2 + bx + c = 0$

Hệ số a

Hệ số b

Hệ số c

Phương trình có hai nghiệm $x_1 = 1.0$ và $x_2 = 0.4$

b) Tính chỉ số BMI. Biết công thức tính chỉ số BMI như sau:

$$\text{BMI} = \text{cân-nặng (kg)} / (\text{chiều-cao (m)} * \text{chiều-cao (m)})$$

Chỉ số BMI được dùng đánh giá trọng lượng cơ thể cho người Châu Á như sau:

- Thiếu cân $\text{BMI} < 18.5$
- Bình thường $18.5 \leq \text{BMI} < 25$
- Thừa cân $25 \leq \text{BMI} < 30$
- Béo phì $\text{BMI} \geq 30$

Tính chỉ số BMI

Cân nặng (kg)

Chiều cao (m)

Tính chỉ số BMI

Cân nặng (kg)

Chiều cao (m)

BMI = 22.49: Bình thường

c) Tính lãi suất tiết kiệm ngân hàng. Biết công thức tính lãi suất ngân hàng như sau:

$$\text{Tiền lãi} = (\text{kỳ-hạn} * \text{lãi-suất} * \text{số-tiền}) / (12 * 100)$$

Tính lãi suất tiết kiệm

Số tiền (VNĐ)

Kỳ hạn (tháng)

Lãi suất (%/năm)

Tính

Tính lãi suất tiết kiệm

Số tiền (VNĐ)

Kỳ hạn (tháng)

Lãi suất (%/năm)

Tiền lãi: 6500000.00 VNĐ

Tính

d) Đổi đơn vị tiền tệ từ VNĐ sang USD, EUR, và AUD. Giả sử rằng:

- 1 EUR = 26.000 VNĐ
- 1 USD = 22.000 VNĐ
- 1 AUD = 16.000 VNĐ

Chuyển đổi đơn vị tiền tệ

Số tiền (VNĐ)

Đơn vị đổi

Đổi

Chuyển đổi đơn vị tiền tệ

Số tiền (VNĐ)

Đơn vị đổi

20000000VNĐ = 909.09USD

Đổi

Trong mỗi ứng dụng, các ô nhập đều là bắt buộc. Do đó, khi người dùng click vào button tính toán tương ứng, hãy thực hiện kiểm tra dữ liệu các ô có rỗng hay không, nếu có thì tạo một đường biên đỏ ẩn/hiện liên tiếp xung quanh ô nhập liệu đó.

Bài 2: Thiết kế HTML và viết chương trình javascript xử lý các chức năng sau:

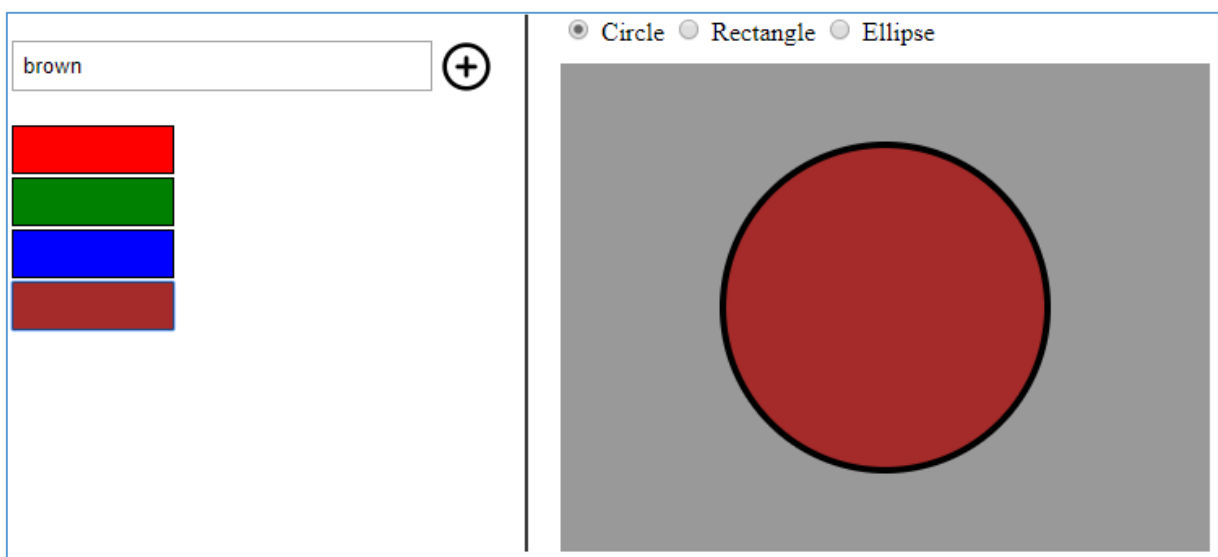
+

☐ Circle
 ☐ Rectangle
 ☒ Ellipse

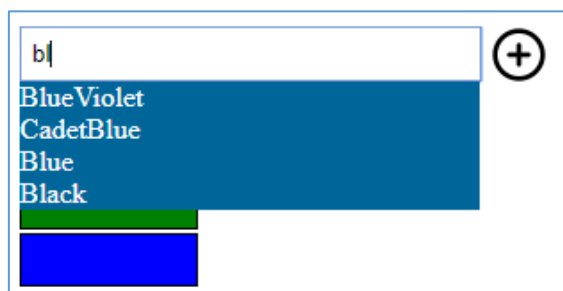
Khi người dùng click chuột trên các radio button (Circle, Rectangle, Ellipse) thì vẽ các hình tương ứng ở dưới bên dưới. Gợi ý sử dụng thẻ `svg` để vẽ các hình này như sau:

```
<svg>
  <circle cx="200" cy="150" r="100" stroke="black"
stroke-width="4" fill="black" />
</svg>
<svg>
  <rect x="100" y="50" width="200" height="200"
stroke="black" stroke-width="4" fill="black" />
</svg>
<svg>
  <ellipse cx="200" cy="150" rx="150" ry="80"
stroke="black" stroke-width="4" fill="black" />
</svg>
```

Khi người dùng click vào các button màu bên trái thì tô màu hình đang được vẽ bên phải với màu tương ứng trên nút bấm. Ngoài ra, người dùng có thể thêm button màu mới bằng cách nhập tên màu hoặc mã màu vào ô nhập trên và click vào button `+` để thêm.



Sau có xử lý thêm javascript: chuẩn bị sẵn một mảng các tên màu và khi người dùng đang nhập màu vào textbox thì tìm kiếm trong mảng những tên màu gần với người dùng nhập và xổ thành một dropdown list để gợi ý nhập cho người dùng.



Bài 3: Thiết kế HTML và tạo các hiệu ứng sau:



Khi trang web vừa nạp lên thì dòng chữ “Welcome to footer!!!” chạy tự động từ phải sang trái và lặp lại vô hạn.

Khi người dùng click trên button “Play” thì bắt đầu thực hiện thao tác tung quả bóng trong vùng màu xám (gray), tức là quả bóng di chuyển trong vùng này, nếu chạm các biên cạnh thì tung ngược lại.

Khi người dùng click trên button “Stop” thì dừng thực hiện tung bóng.

Bài 4: Thiết kế trang web như bên dưới.

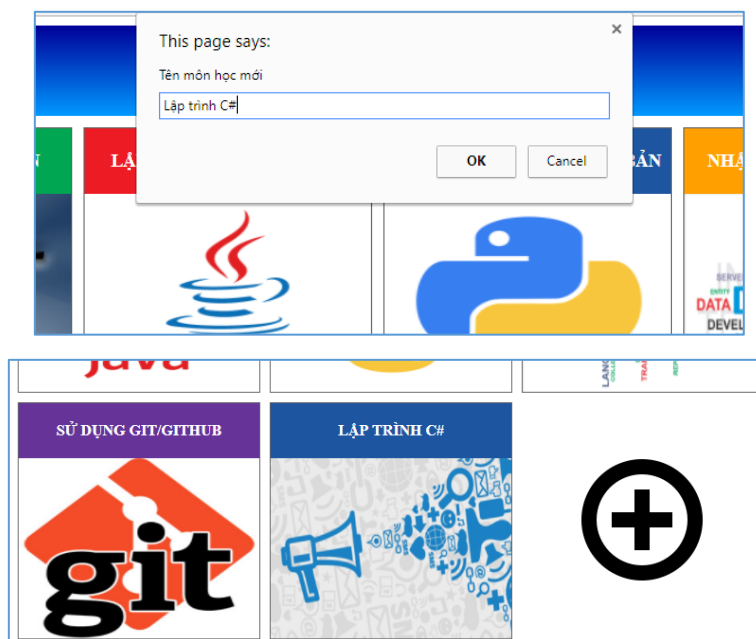


Người dùng có thể click vào button + để thêm một môn học mới thì hiện một hộp thoại dạng prompt để người dùng nhập tên môn học. Trong đó: màu nền của thanh tiêu đề môn học mới tạo lấy ngẫu nhiên từ danh sách màu

[#1D55A0, #ED1B24, #FF9F00, #00A652, #993300, #663399]

Và ảnh cho môn học mới lấy ngẫu nhiên từ danh sách ảnh đính kèm gồm

["category1.jpg", "category2.jpg", "category3.jpg", "category4.jpg", "category5.jpg"]



Khi rê chuột trên từng môn học thì:

- Biên của toàn vùng biên xung quanh hiện lên với bề dày 2px, kiểu solid, màu #666666 và một nút (là siêu liên kết) X xuất hiện góc trên bên phải. Khi người dùng ấn trên nút này thì hiện một hộp thoại dạng confirm hỏi người dùng “*Bạn có chắc chắn muốn xóa môn học này?*”, nếu người dùng click OK thì xóa thành phần của môn học này khỏi HTML.



- Tiêu đề môn học chạy từ phải sang trái, quá trình này lặp lại cho đến khi người dùng rê chuột khỏi vùng môn học.