BÀI 9

JAVASCRIPT
BẮT SỰ KIỆN VÀ TÍNH VỊ TRÍ

PHAN LONG
FB: PHANLONGIT

I. HÀM BẮT SỰ KIỆN

Mục đích: Xác định khi nào người dùng cuộn, rê, clcik chuột

1. Hàm bắt sự kiện cuộn chuột và tính vị trí cuộn chuột:

a. window.addEventListener('scroll',function(){

    các câu lịnh;

})

b. window.pageYOffset: trả về vị trí cuộn chuột theo chiều dọc

Ví dụ:

```
window.addEventListener('scroll',function(){
        //page: trang, Y trục oy, offset vị trí theo trục oy
        console.log(window.pageYOffset);
})
```

c. Đốitượng.offsetTop: Trả về vị trí hiện tại theo chiều dọc của Đối tượng

Bài tập: viết hàm xử lý cho hiệu ứng thay đổi menu khi cuộn chuột (bài 2_buoi21)

Hướng dẫn:

- Bước 1: xử lý phần HTML
- Bước 2: xử lý phần giao diện (css)
- Bước 3: xử lý JS

Áp dụng thực tế:

https://webkhoinghiep.net/theme-wordpress-ban-xe-toyota-01/

II. Viết hiệu ứng slide bằng JavaScript:

Ví dụ: https://themes.muffingroup.com/be/webdesign

https://themes.muffingroup.com/

(Buoi22)

Bước 1: Tạo file HTML

Bước 2: Xử lý nút chuyển trang

Bước 3: Xử lý chuyển slide

Bước 4: Xử lý hiệu ứng

Bước 5: Đặt thời cho slide tự chuyển động

1. Xử lý nút chuyển:

```
<div class="nutchuyen">
    <ul>
        <li class="active">1</li>
        <li>2</li>
        <li>3</li>
    </ul>
</div>
```

```
.nutchuyen{
    text-align: center;
}
.nutchuyen ul li{
    display: inline-block;
    border: 1px solid gray;
    text-indent: -999px;
    width: 20px;
    height: 20px;
    border-radius: 50%;
    margin: 20px 10px;
    transition: 0.5s;
    cursor: pointer;
}
.nutchuyen ul li:hover,
.nutchuyen ul li.active
{    background: gray;
}
```

2. Xử lý slide:

- *Phân tích, đưa các đối tượng vào html*
- *Bố trí vị trí các đối tượng, chia layer*
- *Giới thiệu hàm Sibling: lấy vị trí của đối tượng cùng cấp với đối tượng được click*

* previousElementSibling: vị trí đối tượng trước của đối tượng được click (trả về phần tử trước)

* nextElementSibling: vị trí đối tượng sau của đối tượng được click (trả về phần tử sau)

- *Hàm* setInterval(), setTimeout() : qui định thời gian để làm một công việc nào đó

Hàm Kiểm tra trạng thái của hiệu ứng: webkitAnimationEnd

Kiểm tra khi nào hiệu ứng kết thúc

Cú pháp:

```
Đối tượng.addEventListener('webkitAnimationEnd',function(){
    console.log('Kết thúc chuyển động');
})
```

Ví dụ: B22_testTrangThaiHieuUng

Bước 1: tạo HTML gồm 2 đối tượng

```
<!-- click vào nút thì khối chuyển sang phải -->
    <button class="nut">Click vào nhé</button>
    <div class="khoi"></div>
```

Bước 2: CSS (cho Khối có kích thước và màu)

```
.khoi{
    width: 100px;
    height: 100px;
    background: blue;
}
```

Bước 3: JS thêm class

khi click nút khối chuyển động

```
var nut=document.querySelector('.nut'),
khoi=document.querySelector('.khoi');
nut.addEventListener('click',function(){
    khoi.classList.add('dichuyen');
})
```

Bổ sung class dichuyen

```
.dichuyen{
    animation: abc 1s forwards;
}
@keyframes abc{
  from{transform: translateX(0px);}
  to{transform: translateX(400px);}
}
```

Bước 4: Thêm hàm kiểm tra hành động

```
khoi.addEventListener('webkitAnimationEnd',function(){
    console.log('Kết thúc chuyển động');
})
```

⇒ Console kiểm tra.

Ví dụ: sau khi kết thúc hành động trên bổ sung thêm hành động khác (cho khối đi xuống)

```
khoi.addEventListener('webkitAnimationEnd',function(){
    console.log('Kết thúc chuyển động');
    this.classList.add('dixuong');
})
```

=> Console kiểm tra kết quả

```
.dixuong{
    animation: xuong 1s forwards;
}
@keyframes xuong{
    from{transform: translateX(400px) translateY(0);}
    to{transform: translateX(400px) translateY(300px);}
 }
```