

ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN TIN HỌC ỨNG DỤNG



TÀI LIỆU THỰC HÀNH
NHẬP MÔN LẬP TRÌNH WEB
CT188
(LƯU HÀNH NỘI BỘ)

BIÊN SOẠN: TS. PHẠM TRƯỞNG HỒNG NGÂN



09/2021

BÀI THỰC HÀNH 3: JAVASCRIPT

I. MỤC TIÊU

1. Sử dụng các tính năng cơ bản của Javascript: hàm, ngày giờ
2. Sử dụng Javascript tạo phần tử HTML
3. Xử lý sự kiện các phần tử HTML
4. Xác thực dữ liệu form

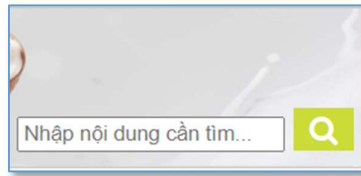
II. LÝ THUYẾT CẦN XEM LẠI

1. Javascript cơ bản
2. DOM HTML
3. Xử lý sự kiện bằng Javascript

III. NỘI DUNG BÀI TẬP


Bài tập 1. Bổ sung chức năng tìm kiếm dữ liệu

Sinh viên hãy thêm chức năng tìm kiếm thông tin cho website bằng cách sử dụng ô tìm kiếm trên đầu trang web (hình 3.1)




Hình 3.1 Ô tìm kiếm ở đầu trang các trang web

Yêu cầu

1. Người dùng sẽ nhập từ khóa tìm kiếm vào ô nhập liệu. Trong quá trình nhập từ khóa, nếu người dùng nhấn phím **enter** hoặc nhấn chuột vào nút tìm kiếm , các sự kiện sẽ được xử lý như sau:
 - Nếu **nội dung ô tìm kiếm khác rỗng**, form tìm kiếm sẽ gửi dữ liệu
 - Nếu **nội dung ô tìm kiếm rỗng**, form tìm kiếm sẽ không gửi dữ liệu
2. Dữ liệu ô tìm kiếm sẽ được gửi đến trang web **timkiem.html**

Phân tích yêu cầu và hướng dẫn thực hiện

1. Trong quá trình nhập từ khóa tìm kiếm
 - a. Người dùng nhấn phím **enter** thì gửi dữ liệu tìm kiếm, do đó cần cài đặt hàm xử lý sự kiện **keypress** của ô nhập liệu. Hàm xử lý sự kiện **keypress** sẽ kiểm tra từng ký tự mà người dùng nhập, nếu **mã** của ký tự là **32** (phím enter) thì tiếp tục gọi thực thi **hàm gửi dữ liệu**.
 - b. Người dùng nhấn chuột vào nút/liên kết  thì gửi dữ liệu tìm kiếm, do đó cần cài đặt xử lý sự kiện **click** của nút/liên kết. Sự kiện **click** của nút/liên kết sẽ được xử lý bằng **hàm gửi dữ liệu**.
 - c. **Hàm gửi dữ liệu:** Hàm sẽ kiểm tra độ dài của ô tìm kiếm, nếu độ dài nội dung của ô lớn hơn 0 thì gửi dữ liệu tìm kiếm, ngược lại thì không gửi.

- Form gửi dữ liệu ô tìm kiếm, nên thuộc tính **method** có giá trị là **GET**, và theo yêu cầu thì thuộc tính **action** của form tìm kiếm có giá trị là **timkiem.html**

Bài tập 2. Xác thực dữ liệu form

Sinh viên hãy bổ sung chức năng xác thực dữ liệu form cho các trang web thành phần có form gửi dữ liệu: trang đăng nhập, đăng ký và liên hệ như hình 3.2

Yêu cầu

- Email có ràng buộc kiểm tra là email hợp lệ
- Mật khẩu phải có độ dài tối thiểu là 8 ký tự
- Họ tên phải có độ dài tối thiểu là 4 ký tự
- Nội dung liên hệ phải có độ dài tối thiểu là 10 ký tự

Phân tích yêu cầu và hướng dẫn thực hiện

Để xác thực dữ liệu form, sinh viên cài đặt trang web theo 2 cách sau:

Cách 1: Sử dụng các tính năng xác thực biểu mẫu tích hợp của HTML5

Bước 1: Thiết lập thuộc tính **onsubmit** của các form và cài đặt hàm xử lý

- Thiết lập thuộc tính **onsubmit** của form là hàm xử lý sự kiện gửi dữ liệu có sử dụng biến con trỏ **this** (chỉ form hiện tại) làm tham số đầu vào như sau:

```
<form onsubmit="return frmValidate5(this);" .....
```

- Hàm xử lý sự kiện **submit** của các form có nội dung là:

```
function frmValidate5(frm)
{
    return frm.checkValidity();
}
```

Bước 2: Bổ sung/ thay đổi thuộc tính các phần tử của form cho phù hợp với ràng buộc dữ liệu như sau:

- Email:** Sử dụng **<input>** thuộc tính **type** là **email**, có thể sử dụng thêm thuộc tính **pattern** với giá trị là biểu thức chính qui xác thực email như sau **^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}**
- Mật khẩu:** Sử dụng **<input>** thuộc tính **type** có giá trị là **password** và thuộc tính **minlength** có giá trị là **8**
- Họ tên:** Sử dụng **<input>** thuộc tính **type** có giá trị là **text** và thuộc tính **minlength** có giá trị là **4**
- Nội dung liên hệ:** Sử dụng **<textarea>** thuộc tính **minlength** có giá trị là **10**

Chú ý: Các ô nhập liệu cần có thuộc tính **required** để có thể xác thực nội dung.

Cách 2: Sử dụng JavaScript lập trình các ràng buộc dữ liệu

- Thiết lập thuộc tính **onsubmit** của form là hàm xử lý sự kiện gửi dữ liệu có sử dụng biến con trỏ **this** (chỉ form hiện tại) làm tham số đầu vào như sau:

```
<form onsubmit="return <tên của hàm xử lý>(this);" .....
```

- Hàm xử lý sự kiện **onsubmit** (tên hàm thay đổi tùy form) có mẫu như sau:

```
function <tên của hàm xử lý>(frm)
{
    /*Các mã kiểm tra ràng buộc dữ liệu*/
    alert("Đã gửi dữ liệu");
    return true;
}
```

- Mã kiểm tra ràng buộc dữ liệu tương ứng với các yêu cầu của bài tập như sau:
 - Email:** Sử dụng biểu thức chính quy và phương thức test để kiểm tra tính hợp lệ của nội dung:

```
var emailReg = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
if ( emailReg.test(<ô nhập liệu email>.value) == false ){
    /*mã xử lý dữ liệu không hợp lệ*/
    return false;
}
```

- Mật khẩu:** Kiểm tra thuộc tính length của nội dung ô nhập mật khẩu

```
if ( <ô nhập liệu mật khẩu>.value.length<8 ) {
    /*mã xử lý dữ liệu không hợp lệ*/
    return false;
}
```

- Họ tên:** Kiểm tra thuộc tính length của nội dung ô nhập họ tên

```
if ( <ô nhập liệu họ tên>.value.length<4 ) {
    /*mã xử lý dữ liệu không hợp lệ*/
    return false;
}
```

- Nội dung liên hệ:** Kiểm tra thuộc tính length của nội dung vùng nhập liệu

```
if ( <vùng nhập liệu nội dung liên hệ>.value.length<10 ) {
    /*mã xử lý dữ liệu không hợp lệ*/
    return false;
}
```

Chú ý: Các thông báo dữ liệu không hợp lệ có thể dùng hàm alert().

Bài tập 3. Thêm sản phẩm vào đơn đặt hàng

Sinh viên hãy bổ sung chức năng đặt hàng cho mỗi sản phẩm trong trang sản phẩm (sanpham.html) như hình 3.2:



Hình 3.2 Chức năng đặt hàng sản phẩm

Yêu cầu

1. Ô nhập liệu số lượng
 - a. Chỉ chấp nhận nhập ký tự số
 - b. Số lượng phải lớn hơn không
 - c. Số lượng tối đa 1 loại sản phẩm trong 1 đơn đặt hàng là 100
2. Nút đặt hàng
 - a. Khi nhấn vào sẽ xác nhận đặt hàng sản phẩm tương ứng
 - b. Có kiểm tra tổng số lượng mỗi sản phẩm đã đặt không vượt quá 100

Phân tích yêu cầu và hướng dẫn thực hiện

Sử dụng danh sách các sản phẩm được mô tả bên dưới để thực hiện yêu cầu

```
var itemList={
  "sp001":{      "name":"Sữa Chua Vị Kiwi",
                  "price":21000,
                  "photo":"images/sanpham/kiwi.jpg"},
  "sp002":{      "name":"Sữa Chua Vị Xoài",
                  "price":22000,
                  "photo":"images/sanpham/mango.jpg"},
  "sp003":{      "name":"Sữa Chua Vị Dưa lưới",
                  "price":23000,
                  "photo":"images/sanpham/cantaloupe.jpg"},
  "sp004":{      "name":"Sữa Chua Vị Mâm Xôi",
                  "price":24000,
                  "photo":"images/sanpham/blackberry.jpg"},
  "sp005":{      "name":"Sữa Chua Vị Dâu Tây",
                  "price":25000,
                  "photo":"images/sanpham/strawberry.jpg"},
  "sp006":{      "name":"Sữa Chua Vị Việt Quất",
                  "price":26000,
                  "photo":"images/sanpham/blueberry.jpg"},
  "sp007":{      "name":"Sữa Chua Vị Bưởi",
                  "price":27000,
                  "photo":"images/sanpham/grapes.jpg"},
  "sp008":{      "name":"Sữa Chua Vị Táo Xanh",
                  "price":28000,
                  "photo":"images/sanpham/green-apple.jpg"},
  "sp009":{      "name":"Sữa Chua Vị Dứa",
                  "price":29000,
                  "photo":"images/sanpham/pineapple.jpg"}
};
```

1. Ô nhập liệu số lượng:
 - a. Chỉ chấp nhận nhập ký tự số: Sử dụng <input> với thuộc tính type có giá trị là number
 - b. Số lượng phải lớn hơn không: thuộc tính min có giá trị là 0
 - c. Số lượng tối đa 1 loại sản phẩm trong 1 đơn đặt hàng là 100: Thuộc tính max có giá trị là 100
2. Nút đặt hàng
 - a. Cần cài đặt hàm xử lý sự kiện **onclick** của nút với tham số đầu vào là **mã sản phẩm tương ứng**

`<button onclick="addCart('code') ">`

* Với **code** là mã sản phẩm tương ứng

b. Hàm xử lý sự kiện thực hiện các việc sau:

- i. Kiểm tra sự tồn tại của mã sản phẩm trong localStorage, nếu không tồn tại thì thêm mới và thiết lập giá trị cho mã sản phẩm

```
if(typeof localStorage[code] === "undefined") {
    window.localStorage.setItem(code, number)
}
```

- ii. Nếu mã sản phẩm đã tồn tại thì kiểm tra tổng của số lượng sản phẩm đã đặt và số lượng sản phẩm đặt mới có vượt quá 100 hay không

- Nếu Tổng vượt quá 100 thì thiết lập số lượng đặt hàng là 100 và hiển thị thông báo

```
window.localStorage.setItem(code, 100)
```

- Nếu Tổng chưa vượt quá 100 thì thiết lập số lượng đặt hàng là tổng của số lượng sản phẩm đã đặt và số lượng sản phẩm đặt mới



```
number=parseInt(document.getElementById(code).value)
current=parseInt(window.localStorage.getItem(code))
window.localStorage.setItem(code, current+number)
```

* Với **code** là mã sản phẩm, **number** là số lượng sản phẩm đặt mới và **current** là số lượng sản phẩm đã đặt.

Bài tập 4. Hiển thị đơn hàng

Sinh viên hãy cài đặt hiển thị đơn hàng theo các yêu cầu sau:

Yêu cầu

1. Khi nhấn vào biểu tượng giỏ hàng  trên phần đầu trang, trình duyệt sẽ chuyển đến trang Đơn hàng (**donhang.html**). Trang Đơn hàng hiển thị nội dung đơn hàng tương tự như hình 3.3 và 3.4
2. Chi tiết đơn hàng được trình bày như sau:
 - a. Tiêu đề nằm ở dòng đầu tiên của bảng
 - b. Chi tiết mỗi sản phẩm trong đơn hàng là 1 dòng trong bảng và hiển thị các thông tin: hình ảnh, tên, số lượng đặt hàng, giá, thành tiền và nút  để xóa đặt hàng sản phẩm, **Giá** và **Thành tiền** có định dạng hiển thị kiểu tiền tệ và đơn vị là Việt Nam Đồng
 - c. Các dòng thông tin cuối đơn hàng:
 - i. **Tổng**: Tổng Thành tiền trước thuế của tất cả sản phẩm trong đơn hàng
 - ii. **Chiết khấu**: Các đơn đặt hàng **đáp ứng cả hai điều kiện sau** sẽ được giảm giá 10% **tổng giá trị đơn hàng trước thuế**
 - Đơn hàng được đặt các ngày thứ 2, thứ 3 và thứ 4
 - Thời gian đặt hàng: Buổi sáng từ 7h sáng đến 11h và Buổi chiều từ 13h đến 17h
 - iii. **Thuế**: thuế VAT, mức thuế là 10% tổng thành tiền đã trừ chiết khấu
 - iv. **Thành tiền**: Tổng trước thuế - chiết khấu + thuế
3. Nếu trang sanpham.html và trang donhang.html được mở trên 2 tab, nếu người dùng cập nhật số lượng đặt hàng của 1 sản phẩm trên trang sanpham.html thì trang donhang.html tự động cập nhật nội dung chi tiết đơn hàng.

The screenshot shows the Freeeze website with a navigation bar containing links: Trang Chủ, Giới Thiệu, Sản Phẩm, Tin Tức, Đăng Nhập, Đăng Ký, and Liên Hệ. A search bar is on the right. The main heading is "CHI TIẾT ĐƠN HÀNG". Below it is a table with 5 columns: Hình SP, Tên SP, Số lượng, Giá, and Thành Tiền. The table is currently empty. To the right of the table, there are calculations for the total order amount: Tổng thành tiền (A) = 0 đ, Chiết Khấu (B) = 0.1 x A = 0 đ, Thuế (C) = 10% x (A - B) = 0 đ, and Tổng đơn hàng = A - B + C = 0 đ. At the bottom of the table is a button "Xác Nhận Đơn Hàng". The footer contains the copyright notice "© 2021 Bản quyền thuộc về FreeezeVN." and social media icons, along with a "TOP" button.

Hình 3.3 Trang Đơn Hàng khi khách hàng chưa đặt hàng sản phẩm

The screenshot shows the Freeeze website with the same navigation bar. The main heading is "CHI TIẾT ĐƠN HÀNG". Below it is a table with 5 columns: Hình SP, Tên SP, Số lượng, Giá, and Thành Tiền. The table contains three rows of data:

Hình SP	Tên SP	Số lượng	Giá	Thành Tiền
	Sữa Chua Vị Dưa lưới	4	23.000 đ	92.000 đ
	Sữa Chua Vị Xoài	4	22.000 đ	88.000 đ
	Sữa Chua Vị Kiwi	4	21.000 đ	84.000 đ

 To the right of the table, there are calculations for the total order amount: Tổng thành tiền (A) = 264.000 đ, Chiết Khấu (B) = 0.1 x A = 26.400 đ, Thuế (C) = 10% x (A - B) = 23.760 đ, and Tổng đơn hàng = A - B + C = 261.360 đ. At the bottom of the table is a button "Xác Nhận Đơn Hàng". The footer contains the copyright notice "© 2021 Bản quyền thuộc về FreeezeVN." and social media icons, along with a "TOP" button.

Hình 3.4 Trang Đơn Hàng hiển thị chi tiết đơn đặt hàng

Phân tích yêu cầu và hướng dẫn thực hiện

- Thiết lập hàm xử lý sự kiện onclick của nút trên phần đầu trang, hàm không có có nội dung như sau:

```
window.location.href = "donhang.html";
```

- Trong trang **donhang.html**, sinh viên tạo bảng và thiết lập nội dung như sau:
 - Dòng tiêu đề của chi tiết đơn hàng được đặt trong thẻ **<thead>**
 - Các dòng Chi tiết mỗi sản phẩm được đặt trong thẻ **<tbody>**

Hàm hiển thị chi tiết đơn hàng (*showCart()*) được thực thi khi trang web được tải hoàn tất có mô tả như sau:

```
TotalPreTax = 0 //Tổng Trước Thuế
Với mỗi key trong danh sách các key của localStorage{
    item = itemList[key] //Thông tin sản phẩm
    photo = item.photo //Hình sản phẩm
    name = item.name //Tên
    price = item.price //Giá
    orderNumber= localStorage.getItem(key) //Số lượng đặt hàng

    //Tạo các ô dữ liệu hiển thị thông tin sản phẩm
    //Tạo ô dữ liệu chứa hình sản phẩm
    Tạo ô dữ liệu để chứa ảnh
    Tạo đối tượng hình ảnh thuộc ô dữ liệu đã tạo ở trên với các thuộc tính
        href = photo
        width = '100px'

    //Tạo các ô dữ liệu hiển thị sản phẩm còn lại
    Tạo các ô dữ liệu có nội dung là name, price và orderNumber
    Tạo ô dữ liệu có nội dung là price x orderNumber
    //Tạo nút xóa
    Tạo thẻ liên kết với các thuộc tính và giá trị như sau:
        href=#
        data-code=key
    Nội dung HTML = <i class="fa fa-trash icon-pink"></i>
    Hàm xử lý sự kiện onclick = removeCart(this.dataset.code)
    Tạo ô dữ liệu và thêm phần tử con là thẻ liên kết đã tạo
    Tạo dòng dữ liệu mới và thêm phần tử con là các ô dữ liệu đã tạo ở trên
    Thêm dòng dữ liệu đã tạo vào <tbody> của bảng
        TotalPreTax = TotalPreTax +(price x orderNumber)
}
Tính các giá trị: chiết khấu, thuế, tổng đơn hàng
Cập nhật nội dung các <span> tương ứng ( thông qua ID ) với kết quả của
chiết khấu, thuế...
```

- Thông tin mỗi sản phẩm (tên, hình, giá bán) lấy từ đối tượng **itemList** ở Bài tập 3
- Số lượng đặt hàng lấy từ localStorage bằng lệnh:
`var number = localStorage.getItem(<mã sản phẩm>)`
- Tạo các ô chứa dữ liệu hình, tên, số lượng, giá bán, thành tiền, nút xóa bằng lệnh: `var <ô dữ liệu> =document.createElement("td")`
- Tạo dòng chứa các ô dữ liệu bằng lệnh:
`var tr= document.createElement("tr");`
- Thêm ô dữ liệu vào 1 dòng bằng lệnh:
`tr.appendChild(<ô dữ liệu>);`
- Tạo đối tượng hình ảnh thuộc một ô dữ liệu bằng lệnh:
`<ô dữ liệu>.innerHTML="<img src='"+ <hình sản phẩm> +' ' class='round-figure' width='100px' />";`

- Hàm xóa sản phẩm khỏi đơn hàng được cài đặt như sau:

```
function removeCart(code)
{
    if(typeof window.localStorage[code] !== "undefined")
    {
        //xóa sản phẩm khỏi localStorage
        window.localStorage.removeItem(code);
        //Xóa nội dung của phần thân của bảng (<tbody>)
        document.getElementById("cartDetail")
            .getElementsByTagName('tbody')[0].innerHTML="";
        //Hiển thị lại nội dung chi tiết của đơn hàng
        showCart();
    }
}
```

* Với **code** là mã sản phẩm

- Định dạng hiển thị kiểu tiền tệ, sử dụng phương thức `Intl.NumberFormat`

c. Các dòng hiển thị kết quả tính toán được đặt trong thẻ **<tfoot>**, có thể dùng các thẻ **** (có đặt ID) để hiển thị các giá trị tính toán như: tổng, thuế,...

i. **Tổng** là giá trị biến `TotalPreTax` trong hàm hiển thị chi tiết đơn hàng

ii. **Tỉ lệ khuyến mãi/chiết khấu (discount)** dựa trên 2 điều kiện

- Điều kiện 1: Sử dụng phương thức `getDay()` của đối tượng `Date()` để lấy thứ của ngày hiện tại
- Điều kiện thứ 2: Giờ đặt hàng (7h sáng đến 11h và 13h đến 17h) có thể chuyển đổi về số phút tương đối trong ngày để cài đặt điều kiện:
 - 7h = 7 x 60 = 420 phút
 - 11h = 11 x 60 = 660 phút
 - 13h = 13 x 60 = 780 phút
 - 17h = 17 x 60 = 1020 phút

- Hàm tính tỉ lệ khuyến mãi/chiết khấu được cài đặt như sau:

```
function getDiscountRate()
{
    var d=new Date();//lấy ngày hiện tại của máy tính
    var weekday=d.getDay();//lấy ngày trong tuần
    var totalMins=d.getHours()*60+d.getMinutes();//đổi
    thời gian hiện tại ra số phút tương đối trong ngày
    if(weekday>=1&&weekday<=3&&((totalMins>=420&&totalMins<=660)||
    (totalMins>=780&&totalMins<=1020)))
        return 0.1;
    return 0;
}
```

iii. **Thuế: tax** = $0.1 \times (\text{TotalPreTax} - \text{discount})$

iv. **Thành tiền** = `TotalPreTax - discount + tax`

3. Để **donhang.html** tự động cập nhật nội dung khi số lượng đặt hàng của 1 sản phẩm trên trang **sanpham.html** thay đổi, sử dụng đoạn mã sau trong tập tin **.js**

```
window.onstorage = () => {
    showCart();
};
```