

ЛАБОРАТОРНАЯ РАБОТА №4

Основы Web-программирования

Задание 1. Сворачиваемы список

Компьютерные сети используются в следующих целях

обеспечение совместного доступа к сетевым ресурсам (данных, оборудования, программ);
обеспечение надёжного хранения данных (в разных местах);
для передачи данных между удалёнными друг от друга пользователями.

Классификация сетей

По возможностям передачи сигналов

узкополосные
широкополосные

По территориальному признаку

по каналам связи

CSS

```
.collapsible {  
  background-color: #777;  
  color: white;  
  cursor: pointer;  
  padding: 18px;  
  width: 100%;  
  border: none;  
  text-align: left;  
  outline: none;  
  font-size: 15px;  
}
```

CSS

```
.active, .collapsible:hover {  
  background-color: #555;  
}
```

```
.collapsible:after {  
  content: '\002B';  
  color: white;  
  font-weight: bold;  
  float: right;  
  margin-left: 5px;  
}
```

CSS

```
.active:after {  
  content: "\2212";  
}
```

```
.content {  
  padding: 0 18px;  
  max-height: 0;  
  overflow: hidden;  
  transition: max-height 0.2s ease-out;  
  background-color: #f1f1f1;  
}
```

HTML

```
<button class="collapsible">Компьютерные сети  
используются в следующих целях</button>
```

```
<div class="content">
```

```
<p>обеспечение совместного доступа к сетевым  
ресурсам (данных, оборудования, программ)</p>
```

```
<p>обеспечение надёжного хранения данных (в  
разных местах)</p>
```

```
<p>для передачи данных между удалёнными друг  
от друга пользователями</p>
```

```
</div>
```

HTML

```
<p>Класификация сетей</p>
```

```
<button class="collapsible">По возможности  
передачи сигналов</button>
```

```
<div class="content">
```

```
<p>узкополосные</p>
```

```
<p>широкополосные</p>
```

```
</div>
```

HTML

```
<button class="collapsible">По  
территориальному признаку</button>
```

```
<div class="content">
```

```
<p>LAN</p>
```

```
<p>MAN</p>
```

```
<p>WAN</p>
```

```
</div>
```


HTML

```
<button class="collapsible">по каналам  
связи</button>
```

```
<div class="content">
```

```
<p>Кабельные</p>
```

```
<p>Радиоканальные</p>
```

```
</div>
```

JS

```
let coll = document.getElementsByClassName("collapsible");
for (let i = 0; i < coll.length; i++) {
  coll[i].addEventListener("click", function() {
    this.classList.toggle("active");
    let content = this.nextElementSibling;
    if (content.style.maxHeight){
      content.style.maxHeight = null;
    } else {
      content.style.maxHeight = content.scrollHeight + "px";
    } }); }
```

Задание

- Изменить цвет раскрывающегося списка
- Изменить цвет заднего фона в раскрытом списке

Компьютерные сети используются в следующих целях

+

Классификация сетей

По возможностям передачи сигналов

—

узкополосные

широкополосные

По территориальному признаку

+

по каналам связи

—

Кабельные

Радиоканальные

Задание 2

Календарь

ПРЕДЫДУЩИЙ

СЛЕДУЮЩИЙ

СЕНТЯБРЬ 2024

Пн	Вт	Ср	Чт	Пт	Сб	Вс
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

CSS

```
html,  
body {  
  height: 100%;  
}  
body {  
  font-size: 100%;  
  line-height: 1.5;  
  font-family: sans-serif;  
  background: #28283b;  
  color: #444;  
}
```

CSS

```
*,
*:before,
*:after {
  box-sizing: border-box;
}

.group:after {
  content: "";
  display: table;
  clear: both;
}
```

CSS

```
img {  
    max-width: 100%;  
    height: auto;  
    vertical-align: baseline;  
}  
  
a {  
    text-decoration: none;  
}
```

CSS

```
.calendar-wrapper {  
  width: 360px;  
  margin: 3em auto;  
  padding: 2em;  
  border: 1px solid #dcdcff;  
  border-radius: 5px;  
  background: #fff;  
}
```


CSS

```
table {  
  clear: both;  
  width: 100%;  
  border: 1px solid #dcdcff;  
  border-radius: 3px;  
  border-collapse: collapse;  
  color: #444;  
}
```

CSS

```
td {  
    height: 48px;  
    text-align: center;  
    vertical-align: middle;  
    border-right: 1px solid #dcdcff;  
    border-top: 1px solid #dcdcff;  
    width: 14.28571429%;  
}
```

```
CSS  td.not-current {
      color: #cococo;
    }
      td.today {
        font-weight: 700;
        color: #28283b;
        font-size: 1.5em;
      }
```

CSS

```
thead td {  
  border: none;  
  color: #28283b;  
  text-transform: uppercase;  
  font-size: 1.5em;  
}
```

CSS

```
#btnPrev {  
  float: left;  
  margin-bottom: 20px;  
}  
  
#btnNext {  
  float: right;  
  margin-bottom: 20px;  
}
```

CSS

```
#btnPrev,  
#btnNext {  
  background: transparent;  
  border: none;  
  outline: none;  
  font-size: 1em;  
  color: #cococo;  
  cursor: pointer;  
  text-transform: uppercase;  
}  
#btnPrev:hover,  
#btnNext:hover {  
  color: #28283b;  
}
```

HTML

```
<div class="calendar-wrapper">  
  <button id="btnPrev"  
    type="button">Предыдущий</button>  
  <button id="btnNext"  
    type="button">Следующий</button>  
  <div id="divCal"></div>  
</div>
```

JS

Дни недели с понедельника

```
let Cal = function(divId) {  
  this.divId  
  this.DaysOfWeek = [  
    'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс' ]; = divId;
```


JS

Месяцы начиная с января

```
this.Months = ['Январь', 'Февраль', 'Март',  
'Апрель', 'Май', 'Июнь',  
'Июль', 'Август', 'Сентябрь',  
'Октябрь', 'Ноябрь', 'Декабрь'];
```

JS

Устанавливаем текущий месяц, год

```
let d = new Date();  
this.currMonth = d.getMonth();  
this.currYear = d.getFullYear();  
this.currDay = d.getDate();  
};
```

JS

Переход к следующему месяцу

```
Cal.prototype.nextMonth = function() {  
  if ( this.currMonth == 11 ) {  
    this.currMonth = 0;  
    this.currYear = this.currYear + 1;  
  }  
  else {  
    this.currMonth = this.currMonth + 1;  
  }  
  this.showcurr();  
};
```

JS

Переход к предыдущему месяцу

```
Cal.prototype.previousMonth = function() {  
  if ( this.currMonth == 0 ) {  
    this.currMonth = 11;  
    this.currYear = this.currYear - 1;  
  }  
  else {  
    this.currMonth = this.currMonth - 1;  
  }  
  this.showcurr();  
};
```

JS

Показать текущий месяц

```
Cal.prototype.showcurr = function() {  
    this.showMonth(this.currYear, this.currMonth);  
};
```

```
Cal.prototype.showMonth = function(y, m) {
```

JS

Показать месяц (год, месяц)

```
let d = new Date()
```

```
, firstDayOfMonth = new Date(y, m, 7).getDay()
```

```
, lastDateOfMonth = new Date(y, m+1, 0).getDate()
```

```
, lastDayOfLastMonth = m == 0 ? new Date(y-1, 11,  
0).getDate() : new Date(y, m, 0).getDate();
```

```
let html = '<table>';
```

JS

Запись выбранного месяца и года

```
html += '<thead><tr>';
```

```
html += '<td colspan="7">' + this.Months[m] + ' ' + y +  
'</td>';
```

```
html += '</tr></thead>';
```

JS

Заголовок дней недели

```
html += '<tr class="days">';  
for(let i=0; i < this.DaysOfWeek.length;i++) {  
    html += '<td>' + this.DaysOfWeek[i] + '</td>';  
}  
html += '</tr>';
```


JS

Записываем дни

```
let i=1;
```

```
do {
```

```
    let dow = new Date(y, m, i).getDay();
```

```
    if ( dow == 1 ) {
```

```
        html += '<tr>';
```

```
    }
```

JS

Если первый день недели не понедельник
показать последние дни предидущего
месяца

```
else if ( i == 1 ) {  
    html += '<tr>';  
    let k = lastDayOfLastMonth - firstDayOfMonth+1;  
    for(let j=0; j < firstDayOfMonth; j++) {  
        html += '<td class="not-current">' + k + '</td>';  
        k++;  
    }  
}
```

JS

Записываем текущий день в цикл

```
let chk = new Date();
let chkY = chk.getFullYear();
let chkM = chk.getMonth();
if (chkY == this.currYear && chkM == this.currMonth && i ==
this.currDay) {
    html += '<td class="today">' + i + '</td>';
} else {
    html += '<td>' + i + '</td>';
}
```

JS

Заккрыть строку в воскресенье

```
if ( dow == 0 ) {  
    html += '</tr>';  
}
```

JS

Если последний день месяца не воскресенье, показать первые дни следующего месяца

```
else if ( i == lastDateOfMonth ) {  
    let k=1;  
    for(dow; dow < 7; dow++) {  
        html += '<td class="not-current">' + k + '</td>';  
        k++;  
    }  
    i++;  
}while(i <= lastDateOfMonth);
```

JS

Записываем HTML в div

```
html += '</table>';
```

```
document.getElementById(this.divId).innerHTML =  
html;
```

```
};
```

JS

Инициализация

```
window.onload = function() {  
  let c = new Cal("divCal");  
  c.showcurr();  
  getId('btnNext').onclick = function() {  
    c.nextMonth(); };  
  getId('btnPrev').onclick = function() {  
    c.previousMonth(); };}  
function getId(id) {  
  return document.getElementById(id);}
```

Задание

- Закрасить текущий день в календаре цветом
- Выделить воскресенье

ПРЕДЫДУЩИЙ

СЛЕДУЮЩИЙ

СЕНТЯБРЬ 2024

Пн	Вт	Ср	Чт	Пт	Сб	Вс
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Задание 3. Список лабораторных работ

Лист заданий		
Ввести текст.....		Добавить задание
✓ Сделать ЛР1		×
✓ Сделать ЛР2		×
✓ Сделать ЛР3		×
Сделать ЛР4		×
Сделать ЛР5		×
Сделать ЛР6		×
Сделать ЛР7		×
Сделать ЛР8		×
Сделать ЛР9		×
Сделать ЛР10		×
Сделать ЛР11		×
Сделать ЛР12		×
Сделать ЛР13		×
Сделать ЛР14		×
Сделать ЛР15		×
Сделать ЛР16		×

Спасибо за рыбу



CSS

```
body {  
  margin: 0;  
  min-width: 250px;  
}  
* {  
  box-sizing: border-box;  
}  
ul {  
  margin: 0;  
  padding: 0;  
}
```

CSS

```
ul li {  
  cursor: pointer;  
  position: relative;  
  padding: 12px 8px 12px 40px;  
  list-style-type: none;  
  background: #eee;  
  font-size: 18px;  
  transition: 0.2s;  
  -webkit-user-select: none;  
  -moz-user-select: none;  
  -ms-user-select: none;  
  user-select: none;  
}
```

CSS

```
ul li:nth-child(odd) {  
  background: #f9f9f9;  
}
```

```
ul li:hover {  
  background: #ddd;  
}
```

```
ul li.checked {  
  background: #888;  
  color: #fff;  
  text-decoration: line-through;  
}
```

CSS

```
ul li.checked::before {  
  content: " ";  
  position: absolute;  
  border-color: #fff;  
  border-style: solid;  
  border-width: 0 2px 2px 0;  
  top: 10px;  
  left: 16px;  
  transform: rotate(45deg);  
  height: 15px;  
  width: 7px;  
}
```

CSS

```
.close {  
  position: absolute;  
  right: 0;  
  top: 0;  
  padding: 12px 16px 12px 16px;  
}
```

```
.close:hover {  
  background-color: #f44336;  
  color: white;  
}
```

CSS

```
.header{  
  background-color: #aabbcc;  
  padding: 30px 40px;  
  color: white;  
  text-align: center;  
}
```

```
.header:after{  
  content: "";  
  display: table;  
  clear: both;  
}
```


CSS

```
input {  
  margin: 0;  
  border: none;  
  border-radius: 0;  
  width: 75%;  
  padding: 10px;  
  float: left;  
  font-size: 16px;  
}  
  
.addBtn:hover {  
  background-color: #bbb;  
}
```

CSS

```
.addBtn {  
  padding: 10px;  
  width: 25%;  
  background: #d9d9d9;  
  color: #555;  
  float: left;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
  transition: 0.3s;  
  border-radius: 0;  
}
```

CSS

```
.addBtn {  
  padding: 10px;  
  width: 25%;  
  background: #d9d9d9;  
  color: #555;  
  float: left;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
  transition: 0.3s;  
  border-radius: 0;  
}
```

HTML

```
<div id="myDIV" class="header">  
  <h2 style="margin:5px">Лист заданий</h2>  
  <input type="text" id="myInput" placeholder="Ввести текст.....">  
  <span onclick="newElement()" class="addBtn">Добавить задание</span>  
</div>
```

HTML

```
<ul id="myUL">
  <li class="checked">Сделать ЛР1</li>
  <li class="checked">Сделать ЛР2</li>
  <li class="checked">Сделать ЛР3</li>
  <li>Сделать ЛР4</li> <li>Сделать ЛР5</li>
  <li>Сделать ЛР6</li> <li>Сделать ЛР7</li>
  <li>Сделать ЛР8</li> <li>Сделать ЛР9</li>
  <li>Сделать ЛР10</li> <li>Сделать ЛР11</li>
  <li>Сделать ЛР12</li> <li>Сделать ЛР13</li>
  <li>Сделать ЛР14</li> <li>Сделать ЛР15</li>
  <li>Сделать ЛР16</li>
</ul>
```

JS

Создать кнопку закрытие для каждого элемента

```
let myNodelist = document.getElementsByTagName("LI");
for (let i = 0; i < myNodelist.length; i++) {
  let span = document.createElement("SPAN");
  let txt = document.createTextNode("\u00D7");
  span.className = "close";
  span.appendChild(txt);
  myNodelist[i].appendChild(span);
}
```

JS

Добавить «отмеченный» символ при нажатии на элемент

```
let close = document.getElementsByClassName("close");
for (let i = 0; i < close.length; i++) {
  close[i].onclick = function() {
    let div = this.parentElement;
    div.style.display = "none";
  }
}
let list = document.querySelector('ul');
list.addEventListener('click', function(ev) {
  if (ev.target.tagName === 'LI') {
    ev.target.classList.toggle('checked');
  }, false);
```

JS

Создайте новый элемент списка при нажатии на кнопку «Добавить»

```
function newElement() {  
  let li = document.createElement("li");  
  let inputValue = document.getElementById("myInput").value;  
  let t = document.createTextNode(inputValue);  
  li.appendChild(t);  
  if (inputValue === "") {  
    alert("Мое любимое задание, а не твое!");  
  } else {  
    document.getElementById("myUL").appendChild(li);  
  }  
}
```


JS

Создайте новый элемент списка при нажатии на кнопку «Добавить»

```
document.getElementById("myInput").value = "";
```

```
let span = document.createElement("SPAN");  
let txt = document.createTextNode("\u00D7");  
span.className = "close";  
span.appendChild(txt);  
li.appendChild(span);
```

JS

Создайте новый элемент списка при нажатии на кнопку «Добавить»

```
for (let i = 0; i < close.length; i++) {  
  close[i].onclick = function() {  
    let div = this.parentElement;  
    div.style.display = "none";  
  }  
}
```

Спасибо за внимание

