

**Комплект оценочных материалов по дисциплине
«Интернет-программирование»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Какой из следующих методов не изменяет исходный массив?

A) splice()

Б) slice()

В) concat()

Г) Все вышеперечисленные методы изменяют массив

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

2. Какой из следующих методов возвращает новый массив, состоящий из элементов исходного массива, которые удовлетворяют предоставленной функции?

A) map()

Б) filter()

В) reduce()

Г) forEach()

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

3. Как можно преобразовать массив объектов в массив значений определенного свойства?

A) map()

Б) filter()

В) reduce()

Г) slice()

Правильный ответ: А

Компетенции (индикаторы): ПК-2

4. Какой способ доступа к свойству объекта является допустимым:

```
var myObj = new Object(),
```

```
  str = "myString",
```

```
  rand = Math.random(),
```

```
  obj = new Object();
```

A) myObj.type = "Dot syntax";

Б) myObj["date created"] = "String with space";

В) myObj[str] = "String value";

Г) myObj[rand] = "Random Number";

- Д) myObj[obj] = "Object";
- Е) myObj[""] = "empty string";

Правильный ответ: Все

Компетенции (индикаторы): ПК-2

5. Какой параметр является обязательным для метода addEventListener?

- А) Объект события
- Б) Функция обратного вызова
- В) Временная метка
- Г) Имя события

Правильный ответ: Г

Компетенции (индикаторы): ПК-2

6. Какой из следующих примеров правильно добавляет обработчик события на кнопку с идентификатором myButton для обработки клика?

- А) document.getElementById('myButton').addEventListener('click', function() { alert('Clicked!'); });
- Б) myButton.addEventListener('click', function() { alert('Clicked!'); });
- В) addEventListener('click', myButton);
- Г) document.addEventListener('myButton', 'click', function() {});

Правильный ответ: А

Компетенции (индикаторы): ПК-2

7. Какой метод используется для отправки запроса с помощью XMLHttpRequest?

- А) sendRequest()
- Б) send()
- В) execute()
- Г) post()

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

8. Какой метод используется для установки типа запроса (GET, POST и т.д.) и URL перед отправкой?

- А) setRequestHeader()
- Б) open()
- В) initialize()
- Г) configure()

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

Задания закрытого типа на установление соответствия

1. Сопоставьте методы с их описанием:

- | Метод | Описание |
|-----------------------|---|
| 1) XMLHttpRequest | А) Метод для отправки и получения данных с сервера без обновления страницы. |
| 2) JSON | Б) Стандартный формат обмена данными, основанный на JavaScript. |
| 3) DOM | В) Объект, который позволяет взаимодействовать с сервером для асинхронного обмена данными |
| 4) fetch() | Г) Способ получения данных из API, который возвращает промис. |
| 5) Асинхронный запрос | Д) Модель, представляющая структуру документа, позволяет манипулировать элементами HTML |

Правильный ответ:

1	2	3	4	5
В	Б	Д	Г	А

Компетенции (индикаторы): ПК-2

2. Сопоставьте методы массивов с их функциями:

- | Метод | Описание |
|--------------|--|
| 1) push() | А) удаляет первый элемент массива |
| 2) pop() | Б) добавляет элемент в конец массива |
| 3) shift() | В) убирает последний элемент массива |
| 4) unshift() | Г) добавляет элемент в начало массива |
| 5) splice() | Д) удаляет и/или добавляет элементы в произвольную позицию массива |

Правильный ответ:

1	2	3	4	5
Б	В	А	Г	Д

Компетенции (индикаторы): ПК-2

3. Сопоставьте типы данных:

- | Объявление | Типы |
|-----------------------|-----------------|
| 1) { "name": "John" } | А) массивы JSON |

- 2) { "age":30 } Б) числа в JSON
 3) {"employee":{"name":"John", "age":30, "city":"New York" }} В) объекты JSON
 4) {"employees":["John", "Anna", "Peter"]} Г) строки в JSON

Правильный ответ:

1	2	3	4
Г	Б	В	А

Компетенции (индикаторы): ПК-2

4. Сопоставьте методы с их описанием:

- | Метод | Описание |
|----------------------------|---|
| 1) setInterval() | А) используется для оптимизации анимаций в веб-приложениях |
| 2) setTimeout() | Б) используется для отмены таймера |
| 3) requestAnimationFrame() | В) устанавливает таймер, который вызывает указанную функцию один раз после заданной задержки (в миллисекундах) |
| 4) clearTimeout() | Г) устанавливает интервал, который повторяет выполнение указанной функции через заданный промежуток времени (в миллисекундах) |

Правильный ответ:

1	2	3	4
Г	В	А	Б

Компетенции (индикаторы): ПК-2

Задания закрытого типа на установление правильной последовательности

1. Определите верную последовательность работы алгоритма AJAX:

- А) Отправка запроса на веб-сервер, объект XMLHttpRequest.
 Б) Создание объекта XMLHttpRequest.
 В) Возникновение событие на веб-странице (страница загружается, нажата кнопка).
 Г) Вывод ответа с помощью JavaScript.
 Д) Отправка сервером ответа на веб-страницу.
 Е) Обработка запроса сервером.

Правильный ответ: В, Б, А, Е, Д, Г

Компетенции (индикаторы): ПК-2

2. Определите верную последовательность для замены содержимого элемента HTML с id "myElement".

А) Получить элемент с помощью `document.getElementById("myElement")`.

Б) Изменить свойство `innerHTML` или `textContent` полученного элемента.

В) Проверить, существует ли элемент на странице (опционально).

Г) Выбрать новый текст или HTML-код для замены.

Правильный ответ: А, В, Г, Б

Компетенции (индикаторы): ПК-2

Задания открытого типа

Задания открытого типа на дополнение

1. Можно добавить свойство, к ранее определённого типу объекта воспользовавшись специальным свойством _____.

Правильный ответ: `prototype`

Компетенции (индикаторы): ПК-2

2. Данные JSON записываются в виде пар ____ - ____.

Правильный ответ: имя, значение

Компетенции (индикаторы): ПК-2

3. Ключи объектов JSON должны быть _____, а значения должны быть допустимым типом данных JSON.

Правильный ответ: строками

Компетенции (индикаторы): ПК-2

4. Чтобы объект был неизменяемым используют _____.

Правильный ответ: `Object.freeze(obj)`

Компетенции (индикаторы): ПК-2

Задания открытого типа с кратким свободным ответом

1. Как установить заголовок для запроса в `XMLHttpRequest`?

Правильный ответ: `setRequestHeader()`

Компетенции (индикаторы): ПК-2

2. Какой метод можно использовать, чтобы проверить, завершился ли запрос?

Правильный ответ: `readyState`

Компетенции (индикаторы): ПК-2

3. Какой метод JavaScript используется для преобразования объекта в строку JSON?

Правильный ответ: JSON.stringify()

Компетенции (индикаторы): ПК-2

4. Какой метод используется для преобразования строки JSON в объект JavaScript?

Правильный ответ: JSON.parse()

Компетенции (индикаторы): ПК-2

Задания открытого типа с развернутым ответом

1. Создать запрос на сервер первых 10 записей в таблице "Customers". Использовать метод JSON.stringify() для конвертации объекта JavaScript в JSON.

Привести расширенное решение.

Время выполнения – 45 мин.

Ожидаемый результат:

Используем JSON.stringify() для конвертации объекта JavaScript в JSON.

```
obj = { "table":"customers", "limit":10 };
dbParam = JSON.stringify(obj);
xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("demo").innerHTML = this.responseText;
    }
};
xmlhttp.open("GET", "json_demo_db.php?x=" + dbParam, true);
xmlhttp.send();
```

Пояснение

- Определяем объект, содержащий свойство Table и свойство Limit.
- Преобразовываем объект в строку JSON.
- Отправляем запрос в PHP-файл со строкой JSON в качестве параметра.
- Ждем, пока запрос вернется с результатом (как JSON).
- Отображаем результат, полученный из файла PHP.

PHP файл:

```
<?php
```

```
header("Content-Type: application/json; charset=UTF-8");
```

```
$obj = json_decode($_GET["x"], false);
```

```
$conn = new mysqli("myServer", "myUser", "myPassword", "Northwind");
```

```

$result = $conn->query("SELECT name FROM ".$obj->table." LIMIT
".$obj->limit);
$outp = array();
$outp = $result->fetch_all(MYSQLI_ASSOC);

```

```

echo json_encode($outp);
?>

```

Пояснение

- Преобразовываем запрос в объект с помощью функции PHP `json_decode()`.

- Выполняем доступ к базе данных, заполняем массив с запрошенными данными.

- Добавляем массив в объект, и возвращаем объект как JSON с помощью функции `json_encode()`.

Критерии оценивания:

- создание объекта, содержащего свойство `Table` и свойство `Limit`;

- преобразование объекта в строку JSON;

- отправка запроса в PHP-файл со строкой JSON;

- преобразование запроса в объект с помощью функции PHP `json_decode()`;

- выполнение доступа к базе данных;

- заполнение массива с запрошенными данными;

- вывод результата, полученного из файла PHP.

Компетенции (индикаторы): ПК-2

2. Выполнить вывод на веб-страницу информации из XML-файла с помощью AJAX.

Привести расширенное решение.

Время выполнения – 45 мин.

Ожидаемый результат:

Создание XML-файл: "cd_catalog.xml".

```
<CATALOG>
```

```
<CD>
```

```
<TITLE>Empire Burlesque</TITLE>
```

```
<ARTIST>Bob Dylan</ARTIST>
```

```
<COUNTRY>USA</COUNTRY>
```

```
<COMPANY>Columbia</COMPANY>
```

```
<PRICE>10.90</PRICE>
```

```
<YEAR>1985</YEAR>
```

```
</CD>
```

```
<CD>
```

```
<TITLE>Hide your heart</TITLE>
```

```
<ARTIST>Bonnie Tyler</ARTIST>
```

```
<COUNTRY>UK</COUNTRY>
```

```

<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
</CATALOG>
Создание файла HTML с кнопкой запроса.
<!DOCTYPE html>
<html>
<style>
table,th,td {
    border : 1px solid black;
    border-collapse: collapse;
}
th,td {
    padding: 5px;
}
</style>
<body>
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Get my CD
collection</button>
<br><br>
<table id="demo"></table>
<script>
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            myFunction(this);
        }
    };
    xhttp.open("GET", "cd_catalog.xml", true);
    xhttp.send();
}
function myFunction(xml) {
    var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Artist</th><th>Title</th></tr>";
    var x = xmlDoc.getElementsByTagName("CD");
    for (i = 0; i <x.length; i++) {
        table += "<tr><td>" +
            x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
            "</td><td>" +
            x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +

```

```

        "</td></tr>";
    }
    document.getElementById("demo").innerHTML = table;
}
</script>
</body>
</html>

```

Пояснение.

– Когда пользователь нажимает кнопку "получить информацию о CD" выше, выполняется функция loadDoc().

– Функция loadDoc() создает объект XMLHttpRequest, добавляет функцию, выполняемую при готовности ответа сервера, и отправляет запрос на сервер.

– Когда ответ сервера готов, создается таблица HTML, узлы (элементы) извлекаются из XML-файла и, наконец, обновляет элемент "Demo" с HTML-таблицей, заполненной XML-данными:

```

LoadXMLDoc()
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            myFunction(this);
        }
    };
    xhttp.open("GET", "cd_catalog.xml", true);
    xhttp.send();
}
function myFunction(xml) {
    var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Artist</th><th>Title</th></tr>";
    var x = xmlDoc.getElementsByTagName("CD");
    for (i = 0; i <x.length; i++) {
        table += "<tr><td>" +
            x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue +
            "</td><td>" +
            x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue +
            "</td></tr>";
    }
    document.getElementById("demo").innerHTML = table;
}

```

Критерии оценивания:

– создание XML-файл: "cd_catalog.xml";

- создание функции loadDoc(), для обработки события нажатия на кнопку;
 - отправка запроса на сервер;
 - получение ответа от сервера;
 - создание таблицы HTML для отображения ответа с сервера.
- Компетенции (индикаторы): ПК-2

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Интернет-программирование» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.03.03 Прикладная информатика.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

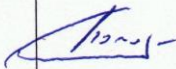
Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии
института компьютерных систем
и информационных технологий



Ветрова Н.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)
1.	Дополнен комплект оценочных материалов	протокол заседания кафедры компьютерных систем и сетей № <u>8</u> от <u>10.03.2025</u>	 С.В. Попов