Создайте верстку:

**HTML**

<h1> Верстка для индивидуальные заданий по <em>jQuery</em> <a href="http://theory.phphtml.net/">на выборку, работу с событиями и анимацией</a> </h1>

<div class="row">

<h2><em>jQuery</em> для начинающих</h2>

<div class="col-md-4">

<br><strong><em>jQuery</em></strong> это замечательный <mark>JavaScript Framework</mark>, который подкупает своей простотой в понимании и удобством в использовании. Но изучение надо с чего-то начинать, и лично моё мнение — лучше всего начинать с наглядных примеров.

</div></div>

<hr color=green>

<div class="row">

<h2>Как получить элемент с помощью <em>jQuery</em>?</h2>

<div class="col-md-4">Для того чтобы понимать как работает <strong>селектор</strong> Вам необходимы базовые знания <mark>CSS</mark> , т.к. именно от принципов <mark>CSS</mark> отталкивает <strong>селектор</strong> <em>jQuery</em> : </div>

<ul>

<li>$("#header") — получение элемента с id=«header»

<li>$(«h3») — получить все &lt;h3&gt; элементы

<li>$(«div#content .photo») — получить все элементы с классом =«photo» которые находятся в элементе div с id=«content»

<li>$(«ul li») — получить все <li> элементы из списка &lt;ul&gt;

<li>$(«ul li:first») — получить только первый элемент &lt;li&gt; из списка &lt;ul&gt;

</ul></div>

<a href="http://theory.phphtml.net/tasks/javascript/jquery/rabota-s-naborom-elementov-jquery.html">Манипулирование элементами страницы</a>

<hr color=green>

<div class="row">

<h2>Выдвижная панель</h2>

<div class="col-md-4">Начнем с простенького <mark>примера</mark> — слайд-панель, она у нас будет двигаться <em>вверх/вниз</em> по клику на ссылке</div>

<div class="col-md-4">Реализуем это следующим образом, по клику на ссылку, у нас будет переключаться её <strong>класс</strong> (между «active» и «btn-slide»), а <strong>панелька </strong>с id=«panel» будет <em>выдвигаться/прятаться</em>. <strong>Класс</strong> «active» изменяет позицию фонового изображения.</div>

<pre>

$(document).ready(function(){

$(".btn-slide").click(function(){

$("#panel").slideToggle("slow");

$(this).toggleClass("active");

});

});

</pre></div>

<a href="http://theory.phphtml.net/tasks/javascript/jquery/rabota-s-sobytiyami-jquery.html">Работа с событиями <em>jQuery</em></a>

<hr color=green>

<div class="row">

<h2>Магические исчезновения</h2>

<div class="col-md-4">Этот <strong>пример</strong> покажет как можно красиво и легко <em>убирать/растворять</em> элементы </div>

<div class="col-md-4">Когда мы <strong>кликаем по картинке</strong> &lt;img class=«delete»&gt;, будет найден <em>родительский элемент</em> &lt;div class=«pane»&gt;,

и его <mark>прозрачность</mark> будет медленно изменяться от opacity= 1.0 до opacity=hide:

</div>

<pre>

$(document).ready(function(){

$(".pane .delete").click(function(){

$(this).parents(".pane").animate({ opacity: "hide" }, "slow");

});

});

</pre></div>

<a href="http://theory.phphtml.net/tasks/javascript/jquery/rabota-s-effektami-i-animaciej-jquery.html">Эффекты и анимация библиотеки <em>jQuery</em></a>

<hr color=green>

**CSS**

\*{font-family: Arial;}

a{color: #369;font-weight: bold;}

.selected{background: #f63;}

.mod{font-size: 2.5em;}

.modyle{color: #669;}

.row {background: grey}

.col-md-4{font-size: 1.2em;}

В документ добавьте 2 кнопки: ЗАДАНИЕ 1 и ЗАДАНИЕ 2. При помощи селекторов jQuery подготовьте документ .js, выполняющий задания соответственно вашего варианта.

## Вариант 1

1. По нажатию на кнопку ЗАДАНИЕ 1, выберите все HTML-элементы **strong** и окрасьте их в зеленый цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который поместит каждый **h2** в **di**v с классом **.mod**.
3. Задайте обработчик событий наведения мышью на элемент **em**: при наведении, цвет элементов должен отличаться от обычного (например, красный), а размер плавно увеличиваться.

## Вариант 2

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы **em** и добавьте им класс **.selected**.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который будет менять цвет элементов с классом **.row**.
3. Задайте обработчик событий наведения мышью на элемент **em**: при наведении, элемент должен плавно перевернуться на 180 градусов. Когда мышь отводим – элемент так же плавно поворачивается в обычное положение.

## Вариант 3

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все элементы **mark**, которые находятся в **div** с классом **.row** и задайте им класс **.selected**.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который перед каждым **h2** вставит **<p>!</p>**.
3. Создайте обработчик нажатия на кнопку СВЕРНУТЬ, который будет сворачивать список.

## Вариант 4

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все гиперссылки и удалите у них подчеркивания.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит цвет всех **li**.
3. Создайте обработчик нажатия на кнопку ПОКАЗАТЬ ЗАГОЛОВКИ, который будет показывать элементы **h2** .

## Вариант 5

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы, которые содержат слово «CSS» и находятся в элементе с классом **. row** и выделите их подчеркиванием.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который будет менять цвет фона страницы.
3. Создайте обработчик нажатия на кнопку ИСЧЕЗНУТЬ, который будет плавно делать список невидимым (прозрачным).

## Вариант 6

1. По нажатию на кнопку ЗАДАНИЕ 1, переключите элементы **em** с классом **.row** в состояние без этого класса, а тем элементам **(em)**, у которых не было этого класса — добавьте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие после **h2**. Если найденный элемент - не **p**, вставьте под заголовком абзац с таким же содержимым, как и заголовок.
3. Задайте обработчик событий наведения мышью на элемент **strong**: при наведении, элемент должен плавно перевернуться на 180 градусов. Когда мышь отводим – элемент так же плавно поворачивается в обычное положение.

## Вариант 7

1. По нажатию на кнопку ЗАДАНИЕ 1, среди набора элементов с классом **.row** удалите класс у второго элемента.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие после **h2**. Если найденный элемент - **div**, добавьте ему в конец содержимое заголовка над ним.
3. Задайте обработчик событий наведения мышью на элемент **strong**: при наведении, цвет элементов должен отличаться от обычного (например, красный), а размер плавно увеличиваться.

## Вариант 8

1. По нажатию на кнопку ЗАДАНИЕ 1, измените CSS-свойство **colo**r у второй гиперссылки в тексте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит размер символов первого элемента, который стоит непосредственно перед элементом с классом **.row.**
3. Создайте обработчик нажатия на кнопку ИСЧЕЗНУТЬ, который будет плавно делать заголовки **h2** невидимыми (прозрачными).

## Вариант 9

1. По нажатию на кнопку ЗАДАНИЕ 1, выберите все HTML-элементы **mark** и окрасьте их в желтый цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который перед каждым **strong** вставит **<b>!!!!</b>**.
3. Задайте обработчик событий наведения мышью на элемент **h2**: при наведении, элемент должен плавно перевернуться на 180 градусов. Когда мышь отводим – элемент так же плавно поворачивается в обычное положение.

## Вариант 10

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы **strong** и добавьте им класс **.mod**.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который будет менять цвет текста страницы.
3. Создайте обработчик нажатия на кнопку ПОСВЕТЛЕТЬ, который будет плавно делать список прозрачным на 50%.

## Вариант 11

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все элементы **mark**, которые находятся в **div** с классом **.col-md-4** и задайте им класс **.mod**.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит цвет первого **li**.
3. Задайте обработчик событий наведения мышью на элемент **h2**: при наведении, элемент должен плавно переместиться вправо на 100px. Когда мышь отводим – элемент так же плавно возвращается в обычное положение.

## Вариант 12

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все гиперссылки и добавьте им заливку желтого цвета.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит размер символов первого элемента, который стоит непосредственно после элемента с классом **.row.**
3. Добавьте любой рисунок. Создайте обработчик нажатия на него, чтобы элемент сжался до размера высота 50px, ширина 100px.

## Вариант 13

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы, которые содержат слово «jQuery» и находятся в элементе с классом **.ro**w и сделайте им заливку на розовом фоне.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие после **h2**. Если найденный элемент - **div**, добавьте ему в начало содержимое заголовка над ним.
3. Создайте обработчик нажатия на кнопку ПОЯВИТЬСЯ, который будет плавно делать список из прозрачного в 100% видимости.

## Вариант 14

1. По нажатию на кнопку ЗАДАНИЕ 1, переключите элементы **stron**g с классом **.row** в состояние без этого класса, а тем элементам (**strong**), у которых не было этого класса — добавьте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие перед **h2**. Если найденный элемент - не **p**, вставьте под заголовком абзац с таким же содержимым, как и заголовок.
3. Задайте обработчик событий наведения мышью на элемент **li**: при наведении, цвет элементов должен отличаться от обычного (например, красный), а размер плавно увеличиваться.

## Вариант 15

1. По нажатию на кнопку ЗАДАНИЕ 1, все заголовки **h2** окрасьте в красный цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит заливку для всех четных **li**.
3. Добавьте любой рисунок. Создайте обработчик нажатия на него, чтобы элемент расширился до размера высота 500px, ширина 1000px.

## Вариант 16

1. По нажатию на кнопку ЗАДАНИЕ 1, измените CSS-свойство **color** у третьей гиперссылки в тексте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие перед **h2**. Если найденный элемент - **div**, добавьте ему в конец содержимое заголовка над ним.
3. Задайте обработчик событий наведения мышью на элемент **li**: при наведении, элемент должен плавно перевернуться на 180 градусов. Когда мышь отводим – элемент так же плавно поворачивается в обычное положение.

## Вариант 17

1. По нажатию на кнопку ЗАДАНИЕ 1, выберите при помощи селекторов jQuery все HTML-элементы **em** и окрасьте их в красный цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит цвет последнего **li**.
3. Создайте обработчик нажатия на кнопку СКРЫТЬ ЗАГОЛОВКИ, который будет скрывать элементы **h2** .

## Вариант 18

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы **em** и добавьте им класс **.modyle**.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие после **li**. Если найденный элемент - не **p**, вставьте под заголовком абзац с таким же содержимым, как и элемент списка.
3. Создайте обработчик нажатия на кнопку ПОСВЕТЛЕТЬ, который будет плавно делать элементы **h2** прозрачными на 50%.

## Вариант 19

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все элементы strong, которые находятся в **div** с классом **.col-md-4** и окрасьте их в желтый цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит цвет с 3-го по 5-й **li**.
3. Задайте обработчик событий наведения мышью на элемент **pre**: при наведении, элемент должен плавно переместиться вправо на 100px. Когда мышь отводим – элемент так же плавно возвращается в обычное положение.

## Вариант 20

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все гиперссылки и замените у них подчеркивания на зачеркивания.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который будет менять цвет элементов с классом **.col-md-4**.
3. Создайте обработчик нажатия на кнопку РАЗВЕРНУТЬ, который будет разворачивать список.

## Вариант 21

1. По нажатию на кнопку ЗАДАНИЕ 1, найдите все HTML-элементы, которые содержат слово «элемент» и выделите их подчеркиванием.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит цвет второго **li**.
3. Добавьте любой рисунок. Создайте обработчик нажатия на него, чтобы элемент вправо на 500px.

## Вариант 22

1. По нажатию на кнопку ЗАДАНИЕ 1, переключите элементы **stron**g с классом **.row** в состояние без этого класса, а тем элементам **(strong)**, у которых не было этого класса — добавьте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит размер символов первого элемента, который стоит непосредственно перед элементом с классом **.col-md-4.**
3. Создайте обработчик нажатия на кнопку СКРЫТЬ, который будет скрывать элементы с гиперссылками.

## Вариант 23

1. По нажатию на кнопку ЗАДАНИЕ 1, все **em**  внутри **h2** окрасьте в красный цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который изменит заливку всех нечетных **li**.
3. Добавьте любой рисунок. Создайте обработчик нажатия на него, чтобы элемент вниз на 500px.

## Вариант 24

1. По нажатию на кнопку ЗАДАНИЕ 1, измените CSS-свойство **color** у третьей гиперссылки в тексте.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие перед **li**. Если найденный элемент - не **p**, вставьте под заголовком абзац с таким же содержимым, как и элемент списка.
3. Задайте обработчик событий наведения мышью на элемент **mark**: при наведении, размер должен плавно увеличиваться. После отведения мыши от элемента, размер его так же плавно должен прийти в норму.

## Вариант 25

1. По нажатию на кнопку ЗАДАНИЕ 1, выберите при помощи селекторов jQuery все HTML-элементы **em** и окрасьте их в фиолетовый цвет.
2. Создайте обработчик нажатия на кнопку ЗАДАНИЕ 2, который находит все элементы непосредственно стоящие перед **h2**. Если найденный элемент - **div**, добавьте ему в начало содержимое заголовка над ним.
3. Создайте обработчик нажатия на кнопку ПОЯВИТЬСЯ, который будет плавно делать список из прозрачного в 100% видимости.