

Всплывающая подсказка
с помощью **html** и **css**



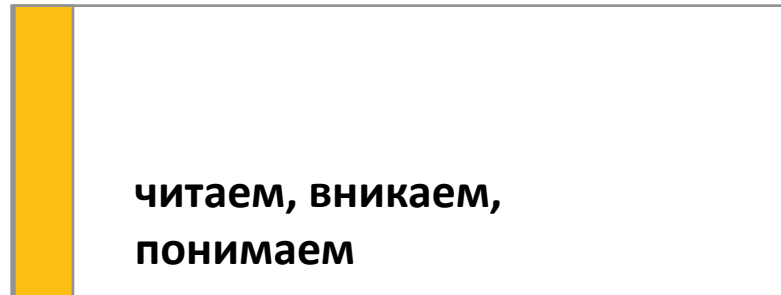
html = hypertext markup language
css = cascade style sheet

ВНИМАНИЕ!

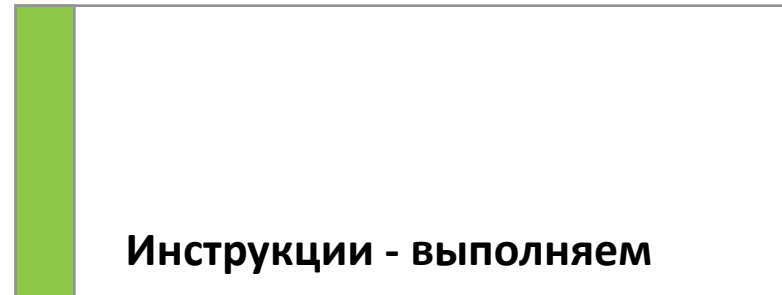
Дальше будут «объяснялки» и инструкции.

Страницы с инструкциями и объяснялками будут **помечены разным цветом**

РЫЖАЯ полоса слева означает,
Что это **«объяснялки»**



ЗЕЛЕНАЯ полоса слева обозначает,
что это **инструкции**



Разметка – в интернете это аналог верстки - делается **тегами**.

Тег обозначает элемент разметки.

Это буква или слово, заключенное между **<** и **>**

Например, тег `<hr>` (*сокращение от **horizontal rule***) – обозначает горизонтальную линию.

Тег `` - шрифт. Тег `` (*image*) – изображение.

Большинство тегов – двойные, состоят из «открывающего» и «закрывающего», между которым находится размечаемый контент:

`Текст написанный этим шрифтом`

(в «закрывающем» теге – обязательный слэш)

Некоторые теги, например `<hr>`, ``, `
` (*break = с новой строки*) – не имеют закрывающего тега.

Наша задача на сегодня:

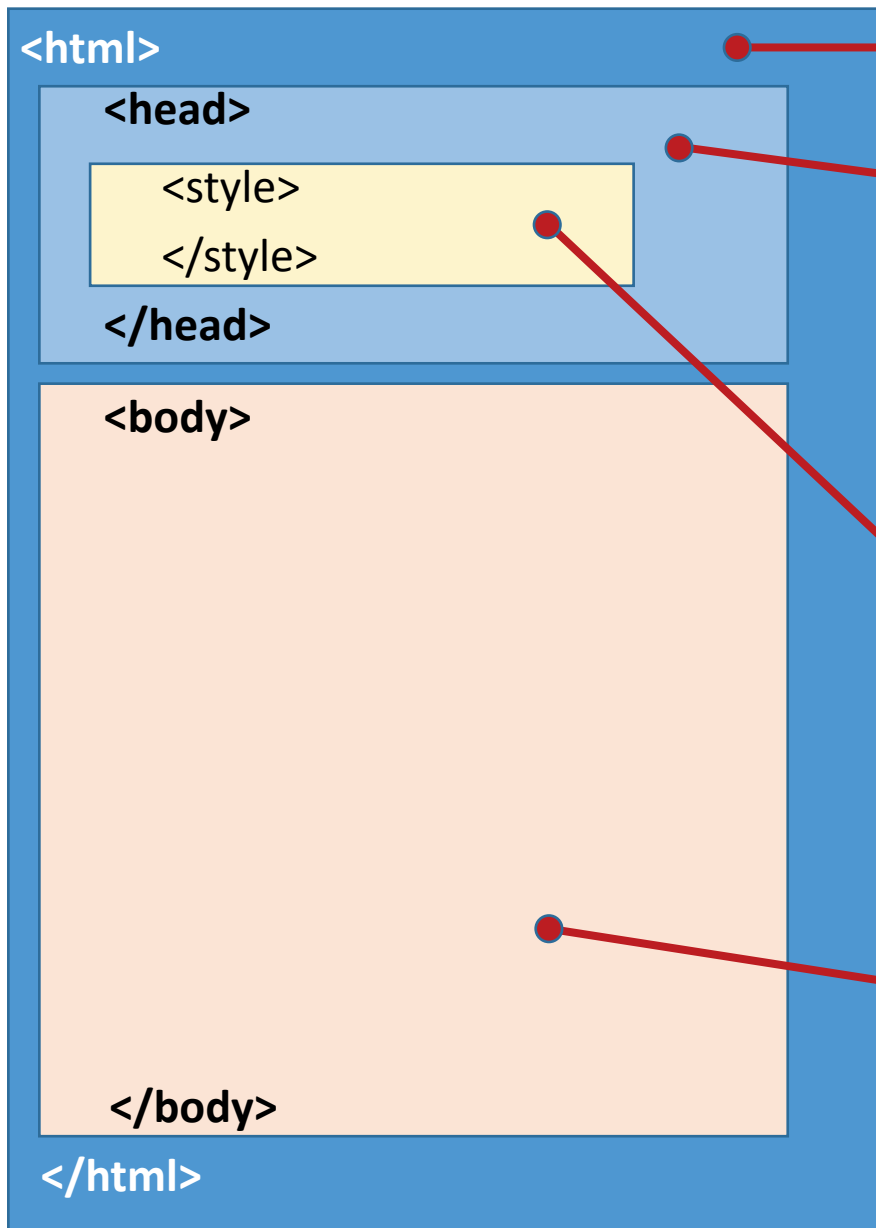
С помощью текстового редактора (Блокнот) и браузера сделать html-страничку с простой всплывающей подсказкой.

Открываем «Блокнот» или «Notepad++»

Начинаем.

HTML-страница размечается так:





<html></html> - общий контейнер

<head></head> - общий контейнер для служебной информации: стилей, скриптов, мета-тегов...
Содержимое тега <head> не отображается напрямую на веб-странице, за исключением тега <title></title> устанавливающего заголовок окна веб-страницы.

<style></style> - место для правил стиля, которые определяют, как выглядит страница и все ее видимые элементы

<body></body> - общий контейнер для ВСЕГО контента, который видим на странице

Вписываем в «Блокнот» структуру страницы:

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body>

  </body>
</html>
```

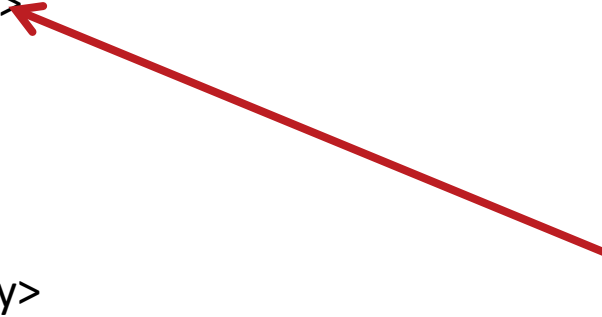
Свойства элементов (цвет, размеры и т.п.)
размечаются в тегах так:

<тег свойство="параметр">

(параметр = в кавычках.

Кавычки – в **английской** раскладке!

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body>
  </body>
</html>
```



Впишем в тег <body> свойство – зеленый цвет страницы:

<Body bgcolor="green">

Bgcolor – background color

Должно получиться так:

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="green">

  </body>
</html>
```

Теперь сохраняем в рабочей папке:

Файл → «сохранить как»

И сохраняем как **page.html**

Внимание: расширение файла после точки пишем обязательно!

Открываем получившуюся страницу в браузере:

Наводим на файл, правая кнопка мыши → Открыть с помощью...

...выбираем Chrome.

Смотрим.

НЕ ЗАКРЫВАЯ БРАУЗЕР, переключаемся в Блокнот.

...и делаем фон нашей страницы белым: `<body bgcolor="white">`

Идем дальше...



```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="white">

  </body>
</html>
```

*Контент располагается между
тегами <body> и </body>*

Теперь, если мы введем в страницу текст, он распределится по всей ширине видимой страницы – читать его будет очень неудобно.

Для удобства – **создадим таблицу**, в которой будет текст и которая ограничит наш текст по ширине....



В разметке таблицы есть три элемента-тега

<table> сама **таблица** **</table>**

В тег таблицы вкладывается:

<tr> = table row (**строка** таблицы) **</tr>**

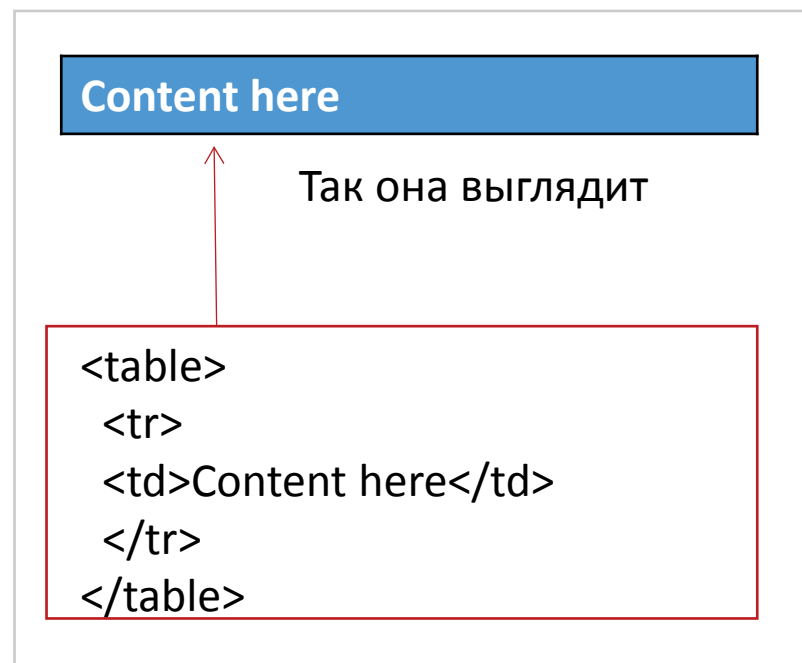
...и в тег строки вкладывается:

<td> = table data (**ячейка** в этой строке и ее содержимое) **</td>**

(обратите внимание: ВСЕ теги таблицы **закрываются**)



Простая таблица из одной ячейки:



(именно такая таблица нам понадобится)

Для справки, разметка таблиц посложнее:

Одна строка, две ячейки

Content 1 here	Content 2 here
----------------	----------------

```
<table>
  <tr>
    <td>Content 1 here</td>
    <td>Content 2 here</td>
  </tr>
</table>
```

Общий принцип:
строка 1 → ячейки слева направо
строка 2 (ниже строки 1) → ячейки слева направо
(и т.д.)

Две строки, две ячейки

Content 1 here	Content 2 here
Content 3 here	Content4 here

```
<table>
  <tr>
    <td>Content 1 here</td>
    <td>Content 2 here</td>
  </tr>
  <tr>
    <td>Content 3 here</td>
    <td>Content4 here</td>
  </tr>
</table>
```



Вставляем простую таблицу внутрь тега <body>

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="white">

</body>
</html>
```

```
<table>
  <tr>
    <td>

  </td>
</tr>
</table>
```

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="white">
  <table>
  <tr>
  <td>

  </td>
  </tr>
  </table>

  </body>
</html>
```

Здесь, внутри таблицы, будет наш текст

Серым цветом обозначены теги, с которыми мы сейчас не работаем...

Серым цветом обозначены теги, с которыми мы сейчас не работаем...

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="white">
<table border="1" width="500">
  <tr>
    <td valign="top">
      место для текста
    </td>
  </tr>
</table>

  </body>
</html>
```

Определяем свойства элементов таблицы:

Граница (border) таблицы: ставим «0» если она невидимая; 1 и больше – толщина видимой линии

Ширина (width) таблицы 500 пикс.

Контент в ячейке по вертикали будет притянут (vertical align = valign) к верхней границе ячейки

Главные теги текстовой разметки:

<h1>Заголовок **</h1>**

<h2>Заголовок **</h2>**

<h3>Заголовок **</h3>**

(*h = headline, заголовок;*
h1 > h2 > h3 по размеру)

<p>Абзац**</p>**

(*p = paragraph*)

**
** С новой строки

(*br = break*)

Заголовок и два абзаца текста будем вписывать между тегами ячейки таблицы **<td>** и **</td>**

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body bgcolor="white">
<table border="1">
  <tr>
    <td valign="top" width="500">
      место для текста
    </td>
  </tr>
</table>

  </body>
</html>
```

Вписываем условные заголовки и два абзаца.
Сразу размечаем:

```
<table border="1">
```

```
<tr>
```

```
<td valign="top" width="500">
```

```
<h1>Заголовок </h1>
```

```
<p>
```

Абзац первый.

```
</p>
```

```
<p>
```

Абзац второй.

```
</p>
```

```
</td>
```

```
</tr>
```

```
</table>
```

Заголовок окружаем тегом заголовка

Начало абзаца: открывающий тег <p>

В конце абзаца – закрывающий тег </p>

*Внутри абзаца можно начинать предложение с новой строки, разделяя тегом
 (такая вот особенность html-разметки)*

СОХРАНЯЕМ наш html
(**ctrl+S** в текстовом редакторе)

Смотрим что получилось:

ПЕРЕКЛЮЧАЕМСЯ В БРАУЗЕР (в нем уже открыт наш файл).

ОБНОВЛЯЕМ его – жмем **F5**

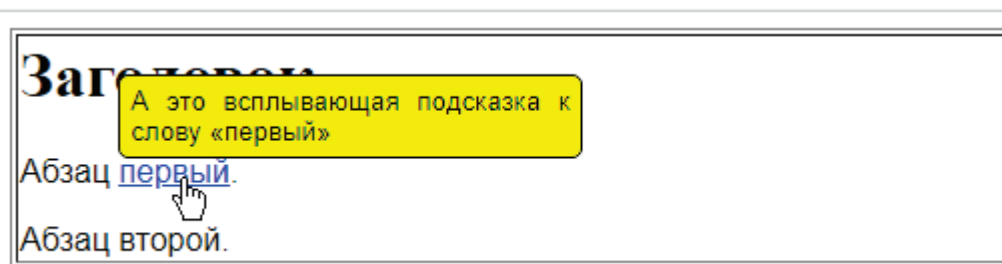
Смотрим что получилось.

НЕ ЗАКРЫВАЯ браузер, возвращаемся в текстовый редактор и продолжаем...

Всплывающая подсказка состоит из:

1. **Триггера** = слова или словосочетания, на которые надо нажать или навести мышку, чтобы появилась
2. **Подсказка** = особым образом оформленный текст, выскакивающий поверх основного текста

Триггером у нас будет работать гиперссылка, которая будет запускать подсказку. Примерно так это должно получиться в конечном итоге:



Вставляем в текст ссылку, по которой будет срабатывать подсказка

```
<table border="1">
  <tr>
    <td valign="top" width="500">
<h1>Заголовок </h1>
<p>
Абзац <a href="#">первый</a>.
</p>
<p>
Абзац второй.
</p>
    </td>
  </tr>
</table>
```

Тег `` ставится ПЕРЕД словом или словосочетанием, которое будет работать ссылкой, в конце ставится закрывающий тег ``

В кавычках после href (hyper reference) вписывают адрес страницы, куда уводит ссылка.

`ТЕКСТ`

«решетка» вместо адреса дает нам ссылку, которая работает внутри этой страницы. Без добавления к «решетке» имени «якоря» (закладки) ссылка работает, но никуда не ведет.

СОХРАНЯЕМ текстовый файл (**ctrl+S**)

Проверка!

Еще раз проходим алгоритм:

1. Сохранив текстовый файл...
2.переходим в браузер
3. Обновляем содержимое страницы в браузере = **жмем F5**
4. Смотрим, что получилось



Сейчас должно получиться что-то вроде такого:

Заголовок
Абзац [первый](#).
Абзац второй.

Не закрываем браузер, возвращаемся в текстовый редактор и продолжаем.


Теперь **текст подсказки** – мы его поместим в контейнер, который появится при нажатии на ссылку.

Тег контейнера: `.....`

Поскольку мы будем жать (или наводить) на ссылку, чтобы появилась подсказка – подсказка зависит от ссылки. То есть контейнер, где будет подсказка станет «дочерним» по отношению к ссылке – или «вложенным» (мы вложим его в тег `<a>`):

```
<p>
Абзац <a href="#"><span>/span>первый</a>.
</p>
```

Структура тегов «вложенного» (дочернего) контейнера:
`<a href>Подсказкаслово-ссылка`



```
<p>
Абзац <a href="#"><span>А это всплывающая подсказка к
слову «первый»</span>первый</a>.
</p>
```

Проверка (начинается с сохранения файла в текстовом редакторе!!):
должно получиться примерно так:

Заголовок

Абзац А это всплывающая подсказка к слову «первый»первый.

Абзац второй.

Так и должно быть.

Потому что дальше мы сделаем подсказку невидимой до тех пор,
пока на нее не наведут мышь...

А потом оформим подсказку: займемся СТИЛЯМИ

Не закрываем браузер и возвращается в текстовый файл...


Теперь нам надо определить, как будет выглядеть и как себя вести наша подсказка.

Это будем определять **СТИЛЯМИ**

```
<html>
  <head>
    <style>

    </style>
  </head>
```

Правила стилей вписываются между тегами **<style> ... </style>** в заголовочной части html-документа



Правила стилей определяются для элементов (тегов) разметки:

Стиль может определять или ВСЕ без исключения элементы (теги),
или только НЕКОТОРЫЕ элементы

```
<style>  
a {правила стиля;}  
  
.hello {правила стиля;}  
</style>
```

Таким способом определяются правила для тега (в данном случае – тега <a>): при такой записи правила будут применены для ВСЕХ тегов <a> в документе.

Таким способом (с точкой перед правилом!) определяются правила для НЕКОТОРЫХ тегов.

Например, для некоторых тегов <a>, когда конкретный тег <a> помечен идентификатором класса тегов, на которые действуют правила стиля:

```
<a href="#" class="hello">
```

(стиль .hello переносится в идентификатор без открывающей точки)

Правило стиля записывается так:

**Правило {
параметр:значение; ← обратите внимание:
}**

- правила пишутся между фигурных скобок
- вместо знака равенства (как в тегах) – двоеточие
- точка с запятой «закрывает» правило

Например:

`body {font:12pt; color:black; font-family:sans-serif;}` = *шрифт во всем документе без засечек, черный, 12 pt*

`a {font:14pt;}` = *кегель шрифта во всех ссылках 14 pt*

`.green {color:green; font:14pt;}` = *стиль для некоторых тегов: шрифт зеленый 14 pt*


Во избежание путаницы:

Когда правила стиля определяются для единичных тегов и самих правил немного – мы можем вставлять их непосредственно в теги, например:

```
<p style="font:14pt blue; align:left;">
```

Но если правила стиля применяются часто, к разным тегам применяются разные правила и этих правил много – тогда, чтобы не загромождать контент и html-разметку, мы отправляем ВСЕ правила ВСЕХ часто применяющихся стилей в head между `<style></style>`

Сейчас считается правильным НЕ определять отдельные правила в отдельных тегах, а выносить правила в `<style></style>` и запускать правила через свойства тегов `class="..."`



Займемся теперь правилами стиля для
подсказки

Важно понять:

контейнер ``, в котором находится подсказка, дочерний: он вложен в тег ссылки:

```
<a href><span>Подсказка</span>слово-ссылка</a>
```

Такой способ (вкладывание) позволяет определять правила стиля для вложенных тегов.

Для вложенного (дочернего) тега действует правило:

Стиль, заданный родительскому тегу, можно передать в дочерний (вложенный) тег.

То есть стиль подсказки мы вложим в тег ссылки:

```
<a href="#" class="стиль подсказки">
```

И так определим этот стиль в `<style></style>`, чтобы на саму ссылку этот стиль не действовал.

Но это еще не все, что нам понадобится...

Стиль может определять «поведение» элемента.

Например, правила стиля определяют, как выглядит ссылка «просто так» и когда на нее наводят мышкой:

```
<style>  
a {color:blue;}      --- стиль исходного состояния ссылки: цвет синий  
a:hover {color:red;} --- hover = «поверх»: курсор при наведении окажется «поверх»  
                       гиперссылки: цвет красный  
</style>
```

ВНИМАНИЕ:

Сначала мы должны определить правила стиля для **исходного состояния** элемента
И только потом – правила стиля для **измененного** состояния

Поэтому у нас будет ТРИ правила стиля для подсказки

1. Правила для подсказки в исходном состоянии

(программисты называют это «объявить правило: у нас есть подсказка»)

2. Правила для подсказки в исходном состоянии как дочерней:

(на правила стиля для подсказки мы укажем «классом» в теге ссылки и...)

(правила стиля, объявленные в теге ссылки, будут переданы в тег с подсказкой);

Главное правило: пока мышь не наведена – подсказка невидима)

3. Правила стиля для дочерней подсказки, когда на ссылку наведена мышка

(правила стиля, объявленные в теге ссылки, уже во 2-м правиле нами переданы в тег);

Главное правило: мышь наведена на ссылку – подсказка появляется с заданным в стиле оформлением



Поехали...

ОБОЗНАЧАЕМ:

1. Правила для подсказки в исходном состоянии:

.tooltip

точка перед названием означает, что стиль будет применен только к тем ссылкам, которые будут запускать подсказки:

``

2. Правила для подсказки в исходном состоянии как дочерней: **.tooltip span**

(правила стиля, объявленные в теге ссылки, будут переданы в тег `` с подсказкой; подсказка невидима)

3. Правила стиля для дочерней подсказки, когда на ссылку наведена мышка: **.tooltip: hover span**

(событие `hover` – на ссылку наведена мышь, - измененные правила передаются в тег `` и подсказка появляется с нужным оформлением)

Вписываем правила в `<style>`, оставляя место для их параметров и значений

```
<head>
  <style>

.tooltip {

}
.tooltip span {

}
.tooltip:hover span {

}

</style>
</head>
```

Вписываем стиль в ссылку,
в которую вложена
подсказка

```
<p>
Абзац первый. Абзац <a href="#" class="tooltip"><span>А это
всплывающая подсказка к слову «первый»</span>первый</a>.
</p>
```

Теперь определяем для правил параметры и их значения – и вписываем их в блок `<style>`:

.tooltip {position:relative; }

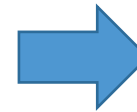
позиционирование: относительное, т.е. точное положение как слова-гиперссылки, так и подсказки определить заранее невозможно;

.tooltip span {display:none;}

исходно подсказка в контейнере невидима, если мышь не наведена;

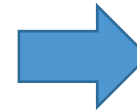
.tooltip:hover span {position:absolute;}

позиционирование: абсолютное, т.е. любая конкретная подсказка будет появляться рядом с конкретной словом-гиперссылкой – в точно определенном относительно любого конкретного слова месте.

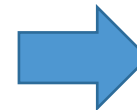


```
<head>
  <style>

.tooltip {
position:relative;
}
```



```
.tooltip span {
display:none;
}
```



```
.tooltip:hover span {
position:absolute;
}

</style>
</head>
```

Сохраняем. Проверяем.

Заголовок

Абзац [первый](#).

Абзац второй.

Теперь подсказка стала невидимой.

Но она невидима, и когда наводим на нее мышкой!

Это нормально: пока мы не определим, как она будет выглядеть, мы ее не увидим.

ИТАК: как будет выглядеть наша подсказка

в правило `.tooltip:hover span` вписываем:

Способ отображения: блок (замкнутая фигура)

`display:block;`

`width:220px;`

`bottom:15px;`

`left:0px;`

Позиционирование:

`width:220px;` (ширина блока)

`bottom:15px;` (расстояние нижней границы блока от верхней границы слова)

`left:0px;` (расстояние левой границы блока от левой границы слова)

`background-color:yellow;`

`border:1px solid black;`

`border-radius:5px;`

Дизайн:

`background-color:yellow;` (заливка – желтая)

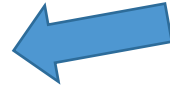
`border:1px solid black;` (граница 1 пикс, сплошная, черная)

`border-radius:5px` (закругление углов)

```
<style>

.tooltip {
position:relative;
}
.tooltip span {
display:none;
}
.tooltip:hover span {
position:absolute;
display:block;
width:220px;
bottom:15px;
left:0px;
background-color:yellow;
border:1px solid black;
border-radius:5px;
}

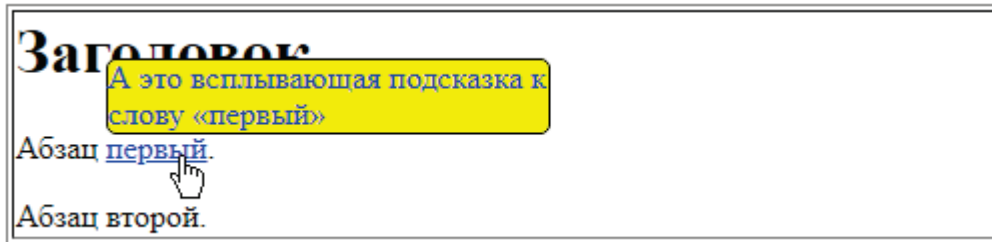
</style>
```



ВПИСЫВАЕМ в .tooltip:hover span

СОХРАНЯЕМ!

ПРОВЕРЯЕМ





В-принципе, главное сделано.

Осталась одна техническая деталь
плюс «навести глянец»

Техническая деталь:

надо сделать, чтобы подсказка всегда выскакивала ПОВЕРХ текста.

Для этого есть специальный параметр **z-index**

Дописываем в **.tooltip** z-index со значением «ноль»;

Дописываем в **.tooltip: hover span** z-index со значением «3»

(в принципе цифры могут быть какие угодно, главное, чтобы значение z-index в .tooltip: hover span было БОЛЬШЕ значения в .tooltip (до поры до времени подсказка все равно будет невидимой))

```
<style>

.tooltip {
  position: relative;
  z-index: 0;
}

.tooltip span {
  display: none;
}

.tooltip: hover span {
  position: absolute;
  display: block;
  width: 220px;
  bottom: 15px;
  left: 0px;
  background-color: yellow;
  border: 1px solid black;
  border-radius: 5px;
  z-index: 3;
}

</style>
```


Наводим «глянец».

Шаг первый:

Определяем шрифт и размер кегля во всех абзацах:

Вписываем в <style> правило:

```
p {  
font: 12pt sans-serif;  
}
```

(то есть ВСЕ абзацы будут напечатаны шрифтом без засечек кеглем 12 pt)

Обратите внимание: 12pt пишется БЕЗ пробела.

А между 10pt и sans-serif пробел ЕСТЬ.

Шаг второй:

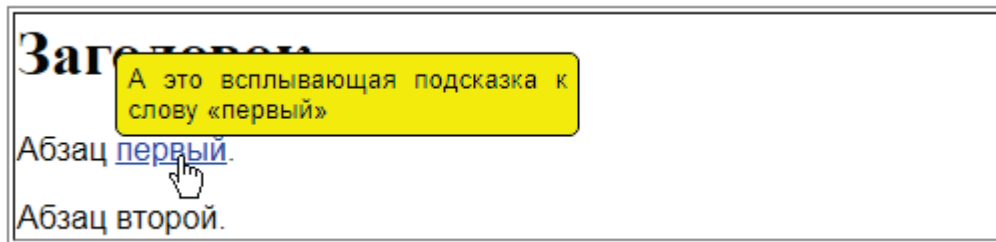
Определяем шрифт и размер кегля в подсказках.
А также еще кое-что из верстки...

Добавляем в `.tooltip:hover span` дополнительные параметры:

<code>padding:5px;</code>	отступы от границ блока в 5 пикс
<code>font:10pt sans-serif;</code>	шрифт без засечек, 10pt (меньше, чем в основном тексте)
<code>color:black;</code>	пусть цвет текста подсказки будет черным
<code>text-decoration:none;</code>	...и без подчеркивания
<code>text-align:justify;</code>	выравниваем текст подсказки по ширине...
<code>text-align-last:left;</code>	... но последняя строка по ширине не растягивается (как в «типографской» верстке)

Для красоты два последних параметра полезно
добавить и в правило абзаца `p {..}`

ПРОВЕРЯЕМ!




Последний штрих: делаем границу таблицы невидимой:
Меняем параметр border на «ноль»:

```
<table border="0">  
  <tr>  
    <td valign="top" width="500">  
      <h1>Заголовок </h1>
```

СОХРАНЯЕМ. ПРОВЕРЯЕМ.





И теперь...

Вписываете в таблицу:

СВОЙ заголовок

СВОЙ текст:

не менее 500 знаков с пробелами, ДВА абзаца,
одно из слов ПЕРВОГО и одно из слов ВТОРОГО абзаца делаете
гиперссылками, запускающими ДВЕ подсказки (текст подсказок – тоже
ВАШ).

Сохраняете готовую страницу в папке «Результаты»

Имя файла: **page_ваша фамилия.html**

И если еще остается время...

Можно сделать дизайн «почти как в Медузе»



Сделаем ссылку непохожей на «традиционную» ссылку.

Обзовем все ссылки, которые с подсказками, например **#tip**

Это будет уникальным именем = «идентификатором» (id),
которым потом пометим нужную ссылку.

Пишем в стилях правила для идентификатора «как есть» и правила для случая,
когда мышку наводят на ссылку

<pre>#tip { background-color:GhostWhite; padding: 2px; border:thin solid grey; border-radius: 3px; text-decoration: none; color:black; }</pre>	<p>фон слова-ссылки серенький; фон окружает текст на 2 пикс; тонкая серая (но потемнее) граница слова-ссылки граница со скругленными на 3 пикс углами Текст ссылки без «подчеркиваний» Шрифт текста черный</p>
<pre>#tip:hover { cursor: default; }</pre>	<p>курсор не меняется при наведении на ссылку (по умолчанию он превращается в «лапку», а нам этого не надо)</p>

Вставляем в тег ссылки наш идентификатор:

```
<a id="tip" href="#" class="tooltip">
```

СОХРАНЯЕМ. ПРОВЕРЯЕМ.

И несколько слов о котиках

Котики - они такие котики.

Будильники, блин. Приходят в пять утра и начинают будить: типа, жрать давай.

Ну ладно, пять утра: уды, воды... Спааать.

Сначала топчутся, потом один начинает вылизывать макушку, а другой искать под одеялом мышь...

А потом они снова приходят будить. Но уже совсем по-другому.

И приходят, заразы такие, вовремя, чтобы в универ успеть.

Котики. Абзац А это всплывающая подсказка к слову «первый»первый.

Спасибо за работу 😊

Сохраните ваш файл в папке «Результаты»))