

ЛЕКЦИЯ. РЕЗИНОВАЯ (ГИБКАЯ) ВЁРСТКА ВЕБ-СТРАНИЦ

Резиновая вёрстка широко применяется в современных веб-проектах и считается базовым профессиональным навыком.

Резиновая вёрстка — это способ построения веб-страницы, при котором размеры элементов задаются в относительных единицах измерения, а не в фиксированных значениях. Основная идея резиновой вёрстки заключается в том, что элементы страницы изменяют свои размеры в зависимости от ширины окна браузера. Ключевым отличием резиновой вёрстки является использование процентов, относительных единиц измерения и гибких контейнеров.

Фиксированная вёрстка предполагает использование строго заданных размеров, например `width: 960px` или `width: 1200px`.

Недостатки фиксированной вёрстки:

- некорректное отображение на экранах малого размера;
- появление горизонтальной прокрутки;
- плохая масштабируемость интерфейса.

Резиновая вёрстка лишена этих недостатков, так как размеры элементов подстраиваются под доступное пространство. Для реализации резиновой вёрстки применяются относительные единицы измерения.

Основные единицы:

- проценты (%);
- `em`;
- `rem`;
- `vw` и `vh`.

Использование относительных единиц позволяет элементам изменять размеры пропорционально размеру экрана или базовому размеру шрифта.

Проценты являются основой резиновой вёрстки. Ширина элемента задаётся относительно родительского контейнера.

Пример:

```
.container {  
  width: 90%;  
  margin: 0 auto;  
}  
  
.block {  
  width: 50%;  
}
```

В данном примере контейнер занимает 90% ширины окна браузера, а вложенный блок — 50% ширины контейнера.

Единица `em` зависит от размера шрифта родительского элемента, а `rem` — от размера шрифта корневого элемента `html`.

Пример:

```
html {  
  font-size: 16px;  
}  
  
.box {  
  width: 20rem;  
  padding: 1.5em;  
}
```

Использование `rem` обеспечивает более предсказуемую масштабируемость интерфейса.

Единицы `vw` и `vh` зависят от размеров окна браузера.

`1vw` равен 1% ширины окна, `1vh` равен 1% высоты окна.

Пример:

```
.hero {  
  width: 100vw;  
  height: 100vh;  
}
```

Такие единицы часто используются для экранных блоков и баннеров.

Flexbox является одним из наиболее удобных инструментов для реализации гибкой вёрстки.

Пример:

```
.container {  
  display: flex;  
}  
  
.item {  
  flex: 1;  
}
```

Все элементы с классом `item` будут автоматически распределены по ширине контейнера.

CSS Grid позволяет строить сложные сетки с гибкими колонками и строками.

Пример:

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 20px;  
}
```

Единица `fr` распределяет доступное пространство между колонками пропорционально.

На практике часто используется смешанный подход, когда часть элементов имеет фиксированные размеры, а часть — гибкие.

Пример:

```
.sidebar {  
  width: 300px;  
}  
  
.content {  
  width: calc(100% - 300px);  
}
```

Данный подход позволяет сохранить стабильность ключевых элементов интерфейса.

Типовые ошибки при резиновой вёрстке

К наиболее распространённым ошибкам относятся:

- использование фиксированных ширин для всех элементов;
- отсутствие ограничения максимальной ширины;
- игнорирование отступов и `box-sizing`;
- неправильное использование процентов.

Избежание этих ошибок является показателем профессионального уровня.