



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) / [6. Адаптивная графика](#) /

📖 6.5. Фавиконки

🕒 ~ 9 минут

Фавиконка — это иконка, которая отображается во вкладке браузера перед названием страницы. Также эта иконка отображается в закладках и на рабочем столе для Веб-приложений.

Подключение

Чтобы иконка начала отображаться, на вкладке нужно подключить файл с графическим изображением иконки. Для правильного подключения иконки нужно использовать обязательные требования браузера и минимальный набор фавиконок, который мы сейчас подробно рассмотрим.

Почему именно набор, а не одну иконку? Дело в том, что браузеры и экраны развиваются и не стоят на месте. Чем современнее браузер, тем более современный формат он может поддерживать. Например, вы можете подумать, что можно подключить только одну svg-иконку и браузер уже сам правильно отрисует её, но не все браузеры до сих пор поддерживают svg-формат для фавиконок. На таблицу совместимости svg-фавиконок можно [посмотреть тут](#).

Обязательная фавиконка

Все — старые и новые — веб-серверы и браузеры поддерживают `.ico`-формат.

Для подключения нужно добавить в `<head>`:

```
<!-- Файл index.html -->

<head>
  <link rel="icon" href="favicon.ico"><!-- 32x32 -->
</head>
```

Обратите внимание на две детали: размер и расположение.

Размер. В интернете можно найти информацию, что нужно подключать фавиконку в формате `.ico` в размере `16×16`. Это так и не так. Дело в том, что контейнер `ico` динамический и он может изменять размеры налету. Поэтому браузеры, которые понимают размер `16×16` сами сожмут `32×32` до необходимого. В каком случае нужно создавать `favicon.ico` в размере `16×16`? Когда фавиконка размером `32×32` при сжатии начнёт мылиться (будет нечёткой). Это можно проверить вручную, сжав фавиконку с `32×32` до `16×16`. Если размыливание картинки произошло, то без дизайнера не обойтись и придётся отрисовывать фавиконку в ручную в графическом редакторе.

Расположение. Фавиконку в формате `ico` обязательно нужно класть в корень проекта, как это показано в примере выше. Дело в том, что любой веб-сервер незаметно для вас всегда ищет `favicon.ico` в корне проекта и пытается её подключить к сайту.

Дополнительные необходимые фавиконки

Расположение. Только `favicon.ico` должна лежать в корне проекта. Остальные иконки могут лежать в любом месте в проекте, например:

```
project/  
  favicon.ico  
  img/  
    180.png  
    192.png  
    512.png  
    icon.svg
```

Или можно сгруппировать все фавиконки, чтобы отделить от других изображений в проекте:

```
project/  
  favicon.ico  
  img/  
    favicons/  
      180.png  
      192.png  
      512.png  
  
  icon.svg
```

SVG

Если современный веб-браузер умеет работать с векторными фавиконками, то этот формат более предпочтительный: меньший вес, наилучшее качество, поддержка

смену тем, не нужно указывать размер.

Подключение точно такое же, как и для `favicon.ico`, только нужно добавить тип, чтобы браузер понимал, с чем имеет дело:

```
<!-- Файл index.html -->

<head>
  <link rel="icon" href="images/favicons/icon.svg" type="image/svg+xml">
</head>
```

Для устройств Apple

Apple пошли по другому сценарию развития и предложили собственные размеры фавиконок для своих устройств — iPhone и iPad. Формат для подключения favicon для Apple поддерживают все современные браузеры.

Самым востребованным размером для старых устройств фавиконок стал размер `180×180`. Давайте посмотрим пример подключения:

```
<!-- Файл index.html -->

<head>
  <link rel="apple-touch-icon" href="images/favicons/apple.png"> <!-- 180×180 -->
</head>
```

Подключив такую иконку, мы также поддержим старые устройства, которые не умеют работать с таким большим размером фавиконки и уменьшать её до нужных размеров.

Устройства Apple работают и с большими размерами иконок, но так получилось, что устройства, которые понимают большие размеры, можно подключать более современным способом с помощью манифеста. Манифест позволит скрыть подключение всех размеров из `<head>`, что сделает разметку более читаемой.

Манифест

Манифест — это JSON-файл со всеми подробностями веб-приложения для браузера. Формат был разработан Google по инициативе [PWA](#).

Чтобы добавить фавиконки с помощью манифеста, нужно подключить манифест, а в манифесте указать все иконки и их размеры, которые мы хотим подключить на сайте.

Файл. Создадим файл в корне проекта и назовём его `manifest.webmanifest`.

Расширение `.webmanifest` нужно указать, чтобы браузер понимал, с каким файлом

имеет дело. Само название файла может быть любым (в нашем случае — `manifest`).

Так как манифест — это JSON-файл, то он должен начинаться с открывающей скобки `{` и заканчиваться закрывающей скобкой `}`.

manifest.webmanifest

```
{  
}
```

Внутри скобок нужно создать секцию иконок. Секция с иконками ожидает несколько иконок, поэтому добавим массив `[]`:

manifest.webmanifest

```
{  
  "icons": []  
}
```

Осталось добавить фавиконки по шаблону:

manifest.webmanifest

```
{  
  "icons": [  
    { "src": "", "type": "", "sizes": "" }  
  ]  
}
```

Если иконок несколько, то их нужно добавить через запятую:

manifest.webmanifest

```
{  
  "icons": [  
    { "src": "", "type": "", "sizes": "" },  
    { "src": "", "type": "", "sizes": "" },  
    { "src": "", "type": "", "sizes": "" }  
  ]  
}
```

Размеры. В манифесте мы укажем размер `192×192` и `512×512`. `192` — так как это следующий размер после `180`, а `512` — так как это очень большой размер для фавиконки, его будет достаточно. А что с промежуточными разрешениями? Всё хорошо: как и раньше браузер сожмёт картинку к нужному размеру. Например, из `512` до `256`.

manifest.webmanifest

```
{
  "icons": [
    { "src": "images/favicons/192.png", "type": "image/png", "sizes": "192x192"
  },
    { "src": "images/favicons/512.png", "type": "image/png", "sizes": "512x512"
  }
  ]
}
```

Настройки:

- `src` — путь до иконки;
- `type` — тип иконки;
- `sizes` — размер иконки.

Можно добавить дополнительные размеры иконок, если этого требует браузерная поддержка вашего проекта. Например, вам потребовалась иконка `256×256`. Давайте добавим её:

manifest.webmanifest

```
{
  "icons": [
    { "src": "images/favicons/192.png", "type": "image/png", "sizes": "192x192"
  },
    { "src": "images/favicons/256.png", "type": "image/png", "sizes": "256x256"
  },
    { "src": "images/favicons/512.png", "type": "image/png", "sizes": "512x512"
  }
  ]
}
```

Откуда брать фавиконки и как их приготовить

Фавиконки — это ответственность верстальщика. Дизайнеры в большинстве случаев про фавиконки не знают, поэтому не предоставляют их с макетами.

Если дизайнер не предоставил фавиконки, у верстальщика есть два пути:

1. Не добавлять фавиконки в проект.
2. Попросить дизайнера отрисовать фавиконку (лучше в векторном формате).
Рассказать, что требуется отрисовать иконку в квадрате. Сама иконка может быть круглой или даже овальной, но изображение должно быть вписано в квадрат с пропорциями 1:1.

Векторный формат позволит растянуть фавиконку до нужных размеров (192, 256 и даже 512) без потери качества. Таким образом верстальщик сможет сгенерировать себе нужные размеры и подключить их к странице.

При генерации фавиконок с помощью сторонних сервисов всегда смотрите на получившийся результат. Чаще всего иконки получаются замыленными. Даже если брать хороший оригинал `512×512`, то при уменьшении до `32×32` всё равно могут появиться артефакты.

Для генерации фавиконок используйте сервис [favicon-generator](#): экспортируйте изображение фавиконки в формате PNG, добавьте это изображение в сервис генерации фавиконок, нажмите кнопку «Create favicon», а потом — ссылку «Download the generated favicon». Вы скачаете архив со всеми возможными вариантами фавиконок. Возьмите `favicon.ico` и другие необходимые версии, подключите к странице проекта.

Итог

Итоговый способ подключения фавиконки:

```
<!-- Файл index.html -->

<link rel="icon" href="favicon.ico"> <!-- 32×32 -->
<link rel="icon" href="images/favicons/icon.svg" type="image/svg+xml">
<link rel="apple-touch-icon" href="images/favicons/apple.png"> <!-- 180×180 -->
<link rel="manifest" href="manifest.webmanifest">
```

manifest.webmanifest

```
{
  "icons": [
    { "src": "images/favicons/192.png", "type": "image/png", "sizes": "192x192"
  },
    { "src": "images/favicons/512.png", "type": "image/png", "sizes": "512x512"
  }
  ]
}
```

Таким способом мы будем поддерживать самые старые браузеры и самые новые

Ничего страшного, что список одной фавиконки получился столь длинным — 4 пункта. Во-первых, браузер загружает фавиконку асинхронно. Во-вторых, он сначала посмотрит на весь список фавиконок и загрузит только одну — самую подходящую.

Прочитали главу?

Нажмите кнопку «Готово», чтобы сохранить прогресс.

Готово

⚠ Если вы обнаружили ошибку или неработающую ссылку, выделите ее и нажмите Ctrl + Enter

Поиск по материалам

Git

[Все материалы](#)

В самом начале



- ☐ Пройдите опрос
- ☐ Укажите персональные данные
- ☐ Изучите регламент
- ☐ Прочитайте FAQ
- ☐ Добавьте свой Гитхаб
- ☐ Выберите наставника
- ☐ Создайте проект

Мой наставник



[Выбрать наставника](#)

Работа с наставником

У вас осталось **10** из 10 консультаций.

[История](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696



Участник

© ООО «Интерактивные обучающие технологии», 2013–2023

