# **Часы - хамелеоны**

# Цель урока

Сделать часы, меняющие цвет в зависимости от времени

# Содержание

1. [HTML/CSS](#_24hqau4jjke)
2. [Что такое классы?](#_yrrpbrcc0i7r)
3. [Получаем дату](#_uaeqnbcs6ftz)

* вывод даты в консоль
* вывод даты на экран
* проверка на длину строки

1. [setInterval](#_p4kfu4ssz7ih)
2. [Программируем кнопку](#_eei3ebwpi87u)
3. [Домашнее задание](#_skt183h2htuk)

# Что повторяем

Синтаксис,объявление переменных, innerHTML, условие if, взаимодействие со стилями с помощью css

# Задачи

* Повторить пройденный материал
* Изучить setInterval
* Познакомиться с классами и this
* Вспомнить getElementById

# Ход урока

У нас будет два элемента, один будет показывать время, а второй текущий цвет фона. Сколько чисел нужно, чтоб указать на текущее время с точностью до секунд? А сколько, чтоб указать цвет? Тоже 6.

## HTML/CSS

|  |
| --- |
| **<!DOCTYPE html> <html> <head>  <title>Clock</title>  <link rel="stylesheet" type="text/css" href="clock.css"> </head> <body>  <h1 id="clock">00:00:00</h1>  <p id="color"></p>  <script src="clock.js"></script>  </body> </html>** |

C CSS будет все предельно просто. Шрифт по больше, отступ сверху и можно еще какой то шрифт импортировать на свой вкус.

|  |
| --- |
| **#clock** {  font-size: 100px; }  **body** {    text-align: center;  font-weight: 800;    margin-top: 250px;  } |

## Что такое классы?

Теперь приступим к логике. Как нам обратиться ко времени в js? Где то оно уже встроено в язык. Есть такая вещь в js, называется “класс”. Это не тот же самый класс, который в html. Это такой шаблон, для создания всякий новых вещей. Например, есть у нас объект - машина, что у всех машин есть общего? Цвет, марка, модель…

Давай научимся создавать класс. Нам нужна переменная, которой мы присвоим этот шаблон. Такие переменные называют с большой буквы, чтоб можно было отличить обычную переменную от класса.

Что мы обычно пишем в функцию в круглых скобках? То, что хотим передать в нее из вне. Нам нужно передавать марку, модель и цвет

|  |
| --- |
| **let** Car = **function**(brand, model, color){   } |

Нам нужно, чтоб мы могли на основе этой функции создавать всякие машины. Если в новую машину передали бренд, модель или цвет, то это свойство является аргументом именно этой машины, а не функции.

Для этого есть такая штука, называется контекст вызова this. Он зависит от того, в каком месте кода мы его используем.

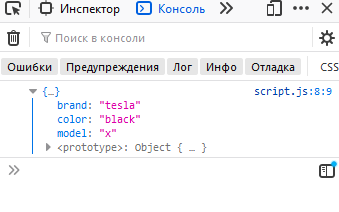
Поэтому, мы должны сказать, что бренд, модель и цвет равны именно тому, откуда их вызывают

|  |
| --- |
| **let** Car = **function**(brand, model, color){  **this**.brand = brand  **this**.model = model  **this**.color = color  } |

Шаблон готов, теперь, создадим саму машину.

|  |
| --- |
| **let** Car = **function**(brand, model, color){  **this**.brand = brand  **this**.model = model  **this**.color = color  } tesla = **new** Car('tesla', 'x', 'black') console.log(tesla); |

Теперь мы можем получать в консоли такие интересные описания и каждая новая машина будет создаваться очень быстро по шаблону.



Создай самостоятельно еще пару машин (для продвинутых создай свой шаблон для чего то еще)

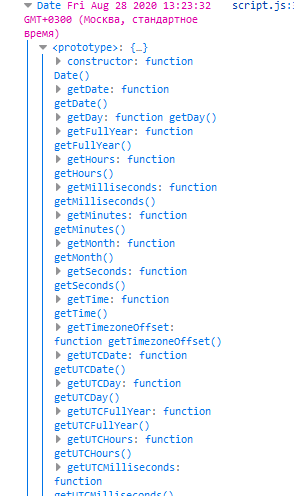
## Получаем дату

А теперь вопрос для размышления, что у нас сидит в каждом из свойств? Строка. А можно там хранить число? А функцию?

В джаваскрипте есть встроенные классы, один из них называется Date. Давай выведем ее в консоль:

|  |
| --- |
| time = **new** Date() console.log(time); |

Смотрим в консоль



Он вернул объект у которого есть prototype. Вообще у каждой функции он есть, но обычно он пустой, а здесь есть куча функций. По названиям можно понять для чего они служат.

У него есть getHours, getMinutes, getSeconds

Посмотрим, что это такое.

|  |
| --- |
| **let** time = **new** Date(); **let** h = time.getHours() **let** m = time.getMinutes() **let** s = time.getSeconds() console.log(h + ':' + m + ':' + s); |



По обновляем страницу, теперь мы выводим время в консоль. Выведем на экран с помощью getElementById

|  |
| --- |
| **let clock = document.getElementById("clock");   let time = new Date(); let h = time.getHours() let m = time.getMinutes() let s = time.getSeconds()  clock.innerHTML = h + ':' + m + ':' + s** |

Ура, при обновлении странички мы видим текущее время.



## setInterval

Теперь нужно заставить вызываться это код раз в секунду. Если нам нужно вызывать код несколько раз - мы используем функцию, а чтоб он вызывался снова и снова раз в секунду, нужно воспользоваться setInterval, в него передается два значения, первое - функция, второе - время в миллисекундах.

|  |
| --- |
| **let** clock = document.getElementById("clock");  **function** **timer**() {  **let** time = **new** Date();  **let** h = time.getHours()  **let** m = time.getMinutes()  **let** s = time.getSeconds()   clock.innerHTML = h + ':' + m + ':' + s } setInterval(timer, 1000) |

Поздравляю, это прогресс! Время обновляется, но после обновления страницы первую секунду мы видим нули, попробуй сам решить эту проблему

|  |
| --- |
| **let** clock = document.getElementById("clock");  **function** **timer**() {  **let** time = **new** Date();  **let** h = time.getHours()  **let** m = time.getMinutes()  **let** s = time.getSeconds()   clock.innerHTML = h + ':' + m + ':' + s } timer() setInterval(timer, 1000) |

Мы говорили, что для обозначения цвета нам тоже нужно использовать 6 цифр, но можем заметить, что иногда у нас их 5, давай если длина строки меньше 2, то будем в начале дописывать ноль. Также нужно их перевести их в строку, так как у нас получится 2 числа вместо 6. Например во времени 20:20:20 их сумма равна 60, а не 202020.

|  |
| --- |
| **let** clock = document.getElementById("clock");  **function** **timer**() {  **let** time = **new** Date();  **let** h = time.getHours().toString();  **let** m = time.getMinutes().toString();  **let** s = time.getSeconds().toString();  **if** (h.length < 2) {  h = "0" + h;  }  **if** (m.length < 2) {  m = "0" + m;  }  **if** (s.length < 2) {  s = "0" + s;  }   clock.innerHTML = h + ':' + m + ':' + s } timer() setInterval(timer, 1000) |

Готово! Теперь нужно привязать цвет фона к этим шести цифрам

|  |
| --- |
| **let** clock = document.getElementById("clock");  **function** **timer**() {  **let** time = **new** Date();  **let** h = time.getHours().toString();  **let** m = time.getMinutes().toString();  **let** s = time.getSeconds().toString();  **if** (h.length < 2) {  h = "0" + h;  }  **if** (m.length < 2) {  m = "0" + m;  }  **if** (s.length < 2) {  s = "0" + s;  }   clock.innerHTML = h + ':' + m + ':' + s  document.body.style.background = '#' + h + m + s } timer() setInterval(timer, 1000) |

Так как у нас самый светлый цвет будет в 23:59:59, поставим цвет фона - белый и наслаждаемся!

Можно усовершенствовать, давай сделаем так, чтоб по кнопке у нас включался и выключался таймер.

Добавим кнопку в и пропишем ей любые стили, на свой вкус.

## Программируем кнопку

|  |
| --- |
| <button type="button" name="button" onclick="start()">On/off</button> |

|  |
| --- |
| **button**{  padding: 20px;  background-color: white;  font-weight: 900  } |

Нужно сделать булеву переменную и две функции.

Включение будет запускать наш setInterval,

|  |
| --- |
| **let** isTimeShowed = false **function** **enableTimer**(){  setInterval(timer, 1000) } |

Функция выключения будет приводить все стили с исходному состоянию и выключать интервал, для это используется clearInterval(), в скобках нужно обратиться к интервалу, который хотим отключить, поэтому, создадим переменную interval в общей области видимости.

|  |
| --- |
| **let** interval **function** **disableTimer**(){  clearInterval(interval)  clock.innerHTML = '00:00:00'  document.body.style.background = 'white'  document.body.style.color = 'black'  }  **function** **enableTimer**(){  interval = setInterval(timer, 1000) } |

А теперь сделаем функцию, которая будет переключать isTimeShowed и если оно true - запускать одну функцию, а иначе - другую

|  |
| --- |
| **function** **start**(){  isTimeShowed=!isTimeShowed  timer()  isTimeShowed ? enableTimer() : disableTimer() } |

Программа готова!

## Домашнее задание

1.Читать, смотреть youtube про setInterval, про классы

2. Сделать цвет текста тоже зависел от времени

Ответ:

(добавить строку типа

|  |
| --- |
| document.body.style.color = '#' + s + 'ffff' |

)

|  |
| --- |
|  |