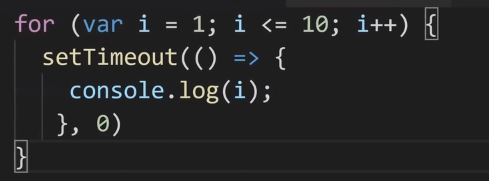
1) Что такое HTML, css, js?  
  
HTML – язык разметки.  
CSS – каскадная таблица стилей.  
JS – язык программирования.  
  
+3) Почему мы не можем вместо тегом section header footer, использовать обычный div?  
- Для скринридеров, поисковиков.  
- Семантика – улучшает доступность сайта.  
  
4) Почему плохо писать все стили в одном файле или style?  
- трудночитаемость  
- потеря логической стуктуры  
- невозможность переиспользования  
  
5) Cascading Style Sheets.  
- Что такое каскад? Иерархия стилей.  
  
6) Как работает наследование CSS?  
- Родительский элемент определяет стили у дочерних.  
  
7) Способы изоляции CSS?  
- Определённые методологии: БЭМ  
  
8) Решает ли БЭМ проблему наследования стилей?  
- Нет. БЭМ не поможет, если у родительского элемента объявлен цвет, то такой же будет у дочерних элементов.  
  
9) Специфичность в CSS.  
.По тегу, классу, айди, по атрибутам или сразу все элементы.  
  
10) Как в CSS работают разные приоритеты стилей?  
Иерархия в CSS.  
  
Приоритет с одинаковой специфичностью – влияет расположения, чем ближе к концу, тем сильнее.  
  
!important > id > class > tag  
  
11) Схлопывание отступов.  
Отступы margin-top и margin-bottom иногда объединяются в один, с размером равным наибольшему из них.  
  
  
  
12) Медиа запросы.  
- Адаптивная вертска под экраны разной ширины.  
- Проверка ширины экрана девайса и подстройка сайта под эту ширину.  
- MobileFirst, desktop First.  
- Адаптивный дизайн и responsive дизайн. Адаптивный – по брейкпоинтам, отзывчивый(responsive) – на любой ширине.  
  
13) Типы данных в JS.  
Number, string, null, undefined, bool, object, bigInt, symbol.  
  
Можно поделить на примитивные и сложные.  
Строки, числа, булин – примитивы, остальное – сложные.  
  
14) Чем похожи массивы, объекты и функции?  
Это всё объекты.  
  
15) Операторы JS.  
Оператор конкатенации – сложение строк и строк, строк и чисел.  
  
16) Операторы сравнения.  
Присваивание, сравнение по значениям, сравнения по типам.  
  
17) Типы написания переменных.  
  
const, let, var.  
Const – нельзя изменить значение переменной.  
let – можно.  
var – можно вызывать до инициализации, всплытие (hoisting).  
  
18) Что такое замыкание?  
Функция, которая может запоминать и ссылаться на внешние переменные.  
  
19)  
   
  
  
  
  
20) Event loop.  
CallStack, 2 очереди.  
  
21) Событийная модель браузеров.  
Фаза погружения, фаза достижения цели, фаза всплытия.  
  
Всплытие событий.  
От цели вплоть до window.  
  
Что бы остановить – e.stopPropagation().