

# Домашние задание #1

## #1 Пройти курс «Знакомство с JavaScript»

Для того, чтобы выполнить задание, нужно полностью пройти курс.

<https://htmlacademy.ru/courses/207>

## #2 Пройти курс «Условия»

Для того, чтобы выполнить задание, нужно полностью пройти курс.

<https://htmlacademy.ru/courses/209>

## #3 Пройти курс «Функции»

Для того, чтобы выполнить задание, нужно полностью пройти курс.

<https://htmlacademy.ru/courses/215>

## #4 Время выбирать

**Задание состоит из нескольких шагов:**

1. Зарегистрируйтесь на [Гитхабе](#).
2. Внесите ссылку на свой Гитхаб в [профиль Академии](#).
3. Выберите [наставника](#) и личный [проект](#), с которым будете работать на интенсиве.
4. Нажмите кнопку «Создать репозиторий» у личного проекта.

После того как вы нажмёте кнопку «Создать репозиторий», наш робот создаст для вас репозиторий проекта в Гитхабе и добавит ссылку на него на [странице выбранного проекта](#).

После выбора наставника и проекта, нажмите зелёную кнопку для проверки и можете переходить к следующему заданию.

## #5 Проверка рабочего процесса

**Задание состоит из нескольких шагов:**

1. Сделайте форк [личного проекта](#). Если в разделе личного проекта нет ссылки на репозиторий, то надо немного подождать, пока наш робот её создаст.
2. Затем клонируйте форк личного проекта себе на компьютер.
3. В форке личного проекта создайте ветку `module1-task1`.
4. В этой ветке внесите изменения в файл `Readme.md`: укажите имя вашего наставника.
5. Сделайте коммиты ваших изменений локально, а затем синхронизируйте локальный и удалённый репозитории (ваш форк).
6. Создайте пулреквест из ветки `module1-task1` вашего форка в мастер-репозиторий и прикрепите задание с помощью кнопки «Найти пулреквест», расположенной над этим заданием в блоке деталей проекта.
7. Нажмите зелёную кнопку «Всё выполнил, можно проверять», расположенную сразу после этого задания.

Если наставник попросит внести дополнительные изменения, то их нужно вносить в форке, коммитить новые изменения и синхронизировать ваш форк. Новый пулреквест создавать не нужно. Все изменения автоматически появятся в уже созданном пулреквесте.

## #6 Учебный проект: начинаем программировать

### Описание учебного проекта

Код и Магия — промо-сайт игры, созданной на основе курса «Двумерные трансформации», с возможностью поиграть в демо-версию, создать вашего уникального персонажа и пройти игру.

### Предыстория

Кекс заказал для своего нового сайта мини-игру с магом и фаерболами. В ней маг может ходить, прыгать и стрелять. Причём управление происходит с помощью стрелок, шифта и пробела, как на компьютере. Действия, которые может осуществлять настоящий маг:

- стрелять фаерболом в случайном направлении (кнопка «шифт»),
- попадать фаерболом в забор и выигрывать,
- подниматься к верхней границе экрана,
- доходить до левого края экрана (стрелка «влево»),
- доходить до правого края экрана (стрелка «вправо»).

Игра была написана и запускается, но Кекс попросил внести несколько модификаций в процесс игры. И эти модификации будете делать вы!

### Подготовка

1. В форке учебного проекта создайте ветку `module1-task1` и в этой ветке выполните следующие шаги.
2. Откройте файл `index.html` в корне учебного проекта.
3. В файле `index.html` добавьте тег `script` непосредственно перед скриптом игры `game.js`.

### Задача

Внутри добавленного тега `script` объявите переменные и функции, которые будут менять поведение игры.

- Объявите переменную `fireballSize`, которая будет управлять размером фаербола в пикселях. Поиграйте размерами фаербола и установите его равным `22`.
- Объявите переменную `getFireballSpeed`, которая будет управлять скоростью фаербола в зависимости от направления ветра в игре. Значением этой переменной должна быть **функция** с одним параметром — `left` (направление полёта фаербола), если параметр `left` — `true`, то фаербол летит по ветру и функция возвращает скорость равную `5`, иначе он летит против ветра со скоростью `2`.
- Установите скорость мага `wizardSpeed` равной `3`.
- Установите ширину мага `wizardWidth` равной `70`.
- Задайте переменной `getWizardHeight` функцию, которая возвращает пропорциональную высоту мага, то есть высота мага — это пропорция `1.337 * ширина мага`.
- Определите функцию `getWizardX` с одним параметром — `width`. Игра будет вызывать вашу функцию и передавать в неё этот параметр (который обозначает ширину игровой области). Результат, возвращаемый вашей функцией, игра будет использовать, чтобы задать положение мага по горизонтали. Напишите функцию так, чтобы центр фигурки мага находился посередине игрового поля.
- Определите функцию `getWizardY` с одним параметром — `height`. Игра будет вызывать вашу функцию и передавать в неё этот параметр (который обозначает высоту игровой области). Результат, возвращаемый вашей функцией, игра будет использовать, чтобы задать положение мага

по вертикали. Напишите функцию так, чтобы маг появлялся на высоте две трети игрового поля от земли.

### **Как тестировать поведение?**

Откройте в браузере `index.html` и начните играть в игру, выполняя описанные выше действия (прыжок до самого верха, выстрел фаерболом и так далее). При изменении переменных в вашем скрипте и перезагрузке страницы в браузере поведение и вид мага будут меняться в соответствии с новыми установками.

### **Обратите внимание**

Координатная сетка игры начинается с левого верхнего угла. То есть точка `[0, 0]` находится слева сверху.

### **Требования**

Код должен соответствовать критериям качества.