# Вступление

Сегодня, благодаря господам спам-ботам и иже с ними, в сети практически невозможно что-либо сделать без ввода символов со сгенерированного изображения — CAPTCHA (далее — капча). Фактически, это предотвращает выполнение какого-либо скрипта без участия человека.

# Основные правила

При разработке капчи нужно обязательно соблюдать несколько основных правил:

**1. Капча создана для людей**

Она должна сходу читаться, но не в ущерб стойкости к распознаванию. Очень наглядный пример капчи, не соблюдающей это правило — изображение ниже.



**2. Генератор капчи должен быть четко ограничен в используемых символах**

Наглядный пример — изображение в начале топика. Конечно, reCAPTCHA — чудесное изобретение, но порой она предлагает ввести символы, которые и в charmap найти сложно. Кстати, когда речь идет о капче с использованием кириллических символов — ни в коем случае генератор не должен использовать букву «ё» (на ~ тильда / ё может быть повешено какое-то действие в системе.

**3. Капча должна быть стойкой к распознаванию**

… но не в ущерб читаемости. Вообще, этот пункт — самый сложный во всей разработке. Необходимо найти золотую середину — капча сходу читается людьми и вообще (насколько возможно) не читается ботами. Так же необходимо учитывать специфику ресурса, на котором планируется использовать капчу.

# Проектирование

Итак, по порядку перечислим, что же мы можем использовать:

**1. В изображении должно использоваться хотя бы несколько цветов.** Желательно всегда разные

https://habrastorage.org/storage/b156bfb2/729b9a93/b5372225/eebe220c.png

Изображение выше — пример того, как это выглядит в действии. На самом деле, это не очень надежный вариант, так как текст с фоном очень контрастирует. Разберемся с цветами позже.

**2. Должен присутствовать шум**

https://habrastorage.org/storage/5291b049/f2e1264e/a7bcb4b3/15dce5a2.png

Прописная истина. Практически в любой капче можно встретить шум, который чаще всего выражается во множестве линий, пересекающих текст, разной длины и под разным углом.

**3. Буквы должны находиться на небольшом расстоянии друг от друга**

https://habrastorage.org/storage/25fa4781/f452acf6/301db087/bda06593.png

Тут главное не перестараться. Чрезмерное сближение символов приведет к сильному ухудшению читаемости человеком. На примере выше можно заметить, что буквы слипаются, это создает препятствие для бота при сегментации изображения.

**4. Размеры символов должны быть разными**

https://habrastorage.org/storage/c2f782be/f7c0f026/3ad45101/d1214ed7.png

Если применять этот трюк, то надо помнить, что препятствие фактически выражается в том, что для сегментации капчи бот не сможет использовать постоянную матрицу. Поэтому если и делать размер символов разным, то обязательно для каждого символа размер должен быть случайным, динамическим.

**5. Отвратительный шрифт**

https://habrastorage.org/storage/e8a95834/3e53b9ae/b2903198/a4eee68e.png

Очень полезный способ. Засечки, курсив, стилизация — отличные подводные камни для бота. Так же в сочетании с шумом в виде линий очень хорошо будет смотреться тонкий шрифт. Если абстрагироваться от первого правила генерации капчи, то можно использовать множество шрифтов сразу, к примеру свой шрифт для каждого символа.

**6. Символы под случайным углом**

https://habrastorage.org/storage/2e03ae15/18de486d/b2f4444f/6e954efb.png

Весьма действенный способ оградиться от ботов. Опять же, сегментация будет усложнена, хоть и не значительно. Лучше всего выбрать небольшой диапазон угла, иначе же читаемость сильно испортится (буквы будут наплывать друг на друга).

**7. Динамические искажения**

https://habrastorage.org/storage/c88b2b5d/7c0251d8/30014b12/1e025f57.gif

Ничего ужаснее человечество еще не придумало. Искажения в капче зачастую очень сильно снижают читаемость человеком. Разумеется, это достаточно эффективно против ботов, но это так же эффективно и против людей. Главное — не перестараться, искажения должны быть незначительными.

**Итак, что же мы сделаем:**

— Контрастный фон, с шумом

— Линии за текстом, линии на тексте

— Текст в случайном положении

— Количество символов будет случайным, от 4 до 7

— Размер каждого символа будет случайным

— Цвет текста каждый раз будет случайным

— Символы будут слегка прикасаться

— Каждый символ будет под случайным небольшим углом

# Разработка

Определимся с целью:

— Генерация шума

— Генерация текста

— Форма с возможностью обновления капчи

— Обработчик введенных данных

# Пишем форму

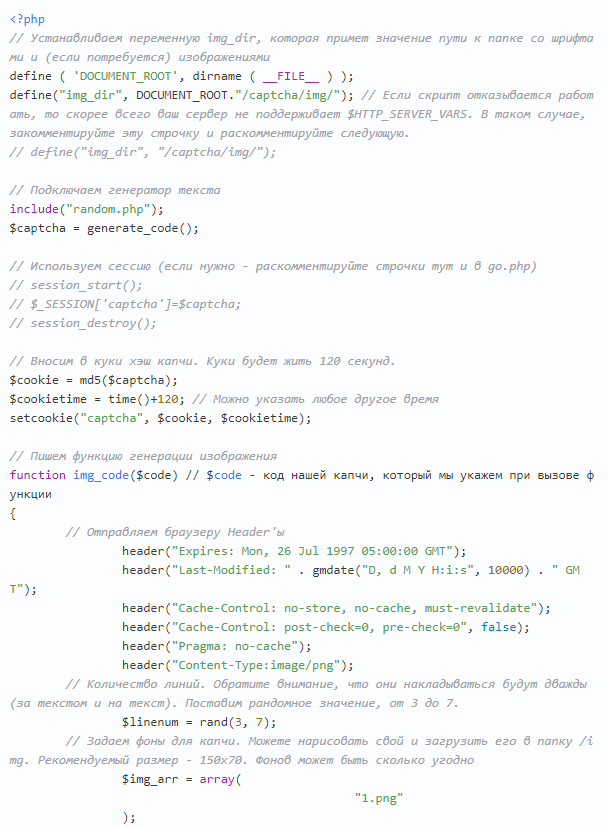


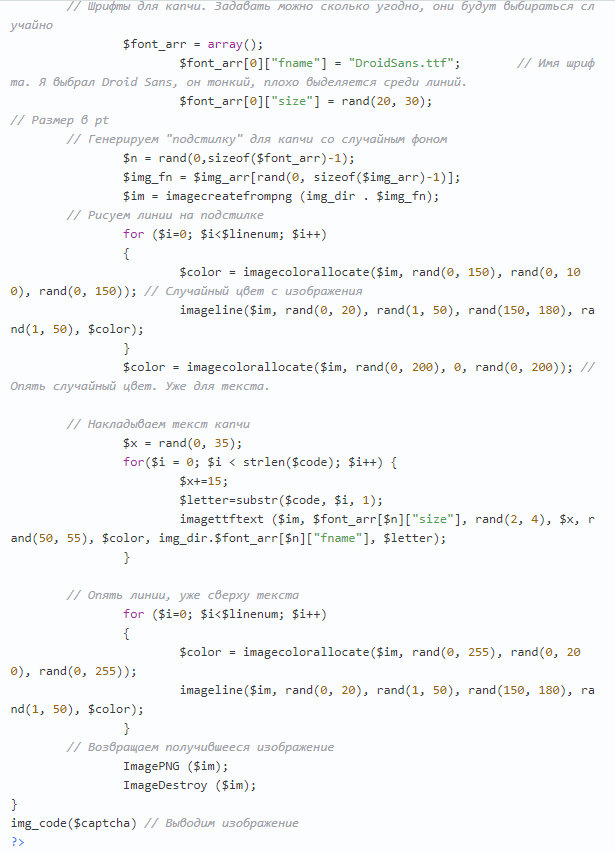
Интерфейс мы написали, пора приступать к созданию скрипта генерации

# Пишем генератор кода капчи (random.php)



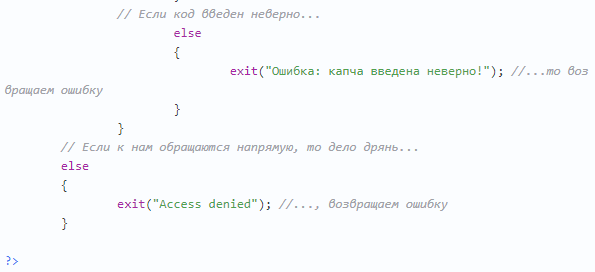
# Пишем генератор изображения (captcha.php)





# Пишем обработчик (go.php)





# Результат

