

УДК 378.621.39 + 004.738.5

ИНТЕРАКТИВНОЕ ВЫПОЛНЕНИЕ ПРОГРАММ НА PYTHON И PHP НА УЧЕБНЫХ САЙТАХ ПЛАТФОРМЫ CSM WORDPRESS

В.В. Костерин, А.В. Зайцев

Использование персональных электронных дневников студентов принесло свои плоды. Система позволила автоматизировано собирать публикации и конспекты студентов, сложней стало с работами студентов в области программирования. Невозможность выполнять лабораторные работы и публиковать их на своем электронном дневнике стало необходимо не только для облегчения образовательного процесса, но и для возможности студентам представить свой электронный дневник в качестве интерактивного резюме для работодателя.

Ключевые слова: Python, WordPress, JavaScript, PHP, web-разработка.

Электронные дневники студентов в формате личных блогов на платформе WordPress [1, 2] позволили вести образовательный процесс в формате виртуального диалога, предоставлять студентам лекционный материал, публикуя его в формате статей на сайте, а так же предоставила возможность студентам публиковать, а преподавателям контролировать, конспекты занятий дистанционных образовательных технологий, рефераты курсовые, научные работы. Такие возможности повышают эффективность образовательного процесса, как при традиционном подходе к обучению, так и при дистанционном [1, 2].

Однако, в плане выполнения практических и лабораторных работ, в частности в курсах, связанных с программированием, приходится использовать сторонние ресурсы, и загружать свои коды в виде файлов или ссылок на такие файлы. Как правило, в качестве такого ресурса используется Github [3]. Этот подход приводит к необходимости создания «распределённого» отчета, когда тексты пояснительных записок располагаются на персональных сайта студентов, а доступ к исполняемому коду можно получить только по ссылке или опубликовать его непосредственно в тексте. Таким образом, чтобы проверить код и поработать над ошибками требуется инструментальное программное обеспечение, которое необходимо дополнительно установить на рабочей станции.

В своей практике мы используем, как минимум, следующие инструментальные пакеты и среды:

1. XAMPP – кроссплатформенная сборка локального веб-сервера, содержащая [Apache](#), [MySQL](#), интерпретатор скриптов [PHP](#), язык программирования [Perl](#) и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер [8].

2. Интерпретатор PHP [9].

3. Интерпретатор скриптов Python [10].

4. IDLE (Integrated Development and Learning Environment) – учебная интегрированная среда для разработки на Python [11] и другие средства разработки.

Работа над кодом значительно осложняется многоэтапностью:

- 1) необходимо написать код,
- 2) отладить его и не запутаться с версиями,
- 3) загрузить его на сторонний ресурс для обмена файлами,
- 4) правильно указать ссылку в своей пояснительно записке на персональном сайте.

Со стороны преподавателя подобная ситуация, так же не отличается удобством. Для проверки кодов в задании, требуется не только прочитать текст пояснений на персональном сайте студента, но и скачать файлы с кодами каждого студента, после чего запустить их с помощью, опять же, перечисленных выше специализированных программ.

Подобную проблемную ситуацию рациональней всего разрешить, расширением возможностей платформы персональных сайтов, в части реализации интерактивного редактора кода с возможностью выполнения кода непосредственно в рамках платформы, т.е. используя только любой браузер для просмотра результата. С таким редактором студенты могут выполнять свои работы прямо в своих электронных дневниках [4] (рис. 1).

После выполнения практической работы, студент может опубликовать её в формате статьи на своем блоге (см. рис. 2).

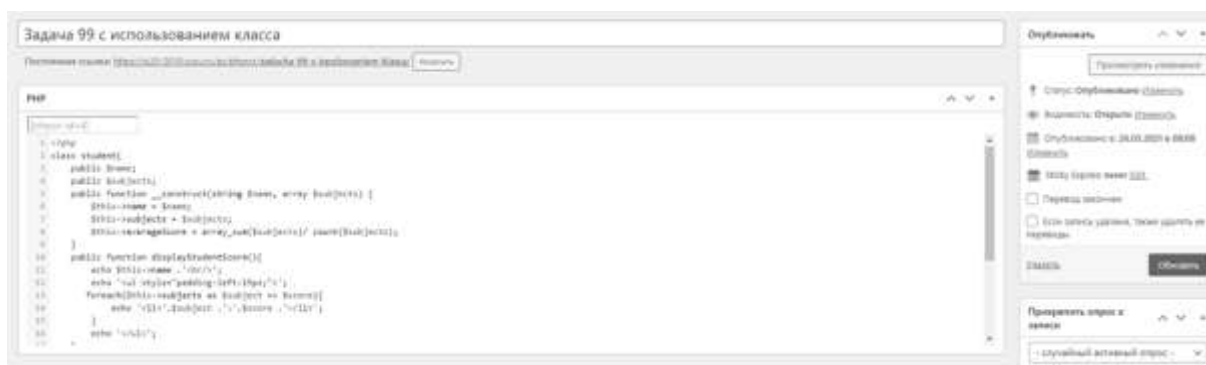


Рис. 5. Интерфейс интерактивного редактора кода

```
1 <?php
2 class student{
3     public $name;
4     public $subjects;
5     public function __construct(string $name, array $subjects) {
6         $this->name = $name;
7         $this->subjects = $subjects;
8         $this->averageScore = array_sum($subjects)/ count($subjects);
9     }
10    public function displayStudentScore(){
11        echo $this->name . '<br/>';
12        echo '<ul style="padding-left:15px;">';
13        foreach($this->subjects as $subject => $score){
14            echo '<li>'.$subject .':'. $score . '</li>';
15        }
16        echo '</ul>';
17    }
18 }
19
```

Results:

Иванов
Математика:3
Русский:4
информатика:5
Средний балл: 4
Петров
Математика:2
Русский:5
информатика:5
Средний балл: 4
Сидоров
Математика:4
Русский:5

Рис. 6. Пример публикации кода и результата его выполнения на персональном сайте студента [4]

Как можно видеть, в публикации приводится не только код, но, и результат его выполнения. Таким образом, при проверке задания, можно открыть публикацию и непосредственно в коде написать замечания, комментарии и советы по исправлению ошибок.

При формулировании заданий студентам, преподаватель может использовать интерактивный редактор для демонстрации примеров кода, и классических приемов, и алгоритмов программирования, так же в формате

публикаций на сайте, доступных студентам в любое время и с любого устройства с выходом в интернет.

Реализация интерактивного редактора для PHP не вызывает особых затруднений. Ядро CMS WordPress [5] поддерживает полный стек web-разработки: PHP, JavaScript, HTML и CSS и поэтому его реализация выполнена без особого вмешательства в платформу в виде плагина. Гораздо сложнее с другими, «не родными» для WordPress, языками программирования, например, Python. Это популярный язык, без которого в современных реалиях нельзя обойтись в социально-экономическом анализе, при решении задач обработки больших данных и искусственного интеллекта.

На текущий момент вопрос запуска скриптов Python в инородной для него web-среде исполнения актуален среди разработчиков. Решение, которое первым приходит в голову и, которое предлагает интернет сообщество – установка интерпретатора python на web-сервере и использование таких функции, как *exec*, *eval*, *system* для запуска и последующего вывода результатов в *iframe*. Но подобное решение годится разве что для вывода строки «Hello world». Не говоря уже о том, что возникает серьезная угроза безопасности, когда в качестве фактических параметров функций *exec*, *system* или *eval* можно передать любой вредоносный код. Конечно, фактические параметры для этих функции можно предварительно тщательно проверять и фильтровать, но все-таки лучшей защитой от злонамеренных инъекций является отказ от использования подобных функций.

Для более качественного решения подобных проблем, стоит обратить внимание, на такой проект как Brython [XX]. Brython – это и компилятор Python, и интерпретатор, написанный на JavaScript. В результате вы можете компилировать и запускать код Python в браузере. Хороший пример этой функции демонстрирует онлайн-редактор, доступный на веб-сайте Brython.

В онлайн-редакторе Python работает в браузере. Нет необходимости устанавливать Python на машину, нет необходимости отправлять код на сервер для выполнения. Пользователь получает немедленную обратную связь, и такой подход не подвергает серверную часть вредоносным скриптам. Студенты могут экспериментировать с Python на любом устройстве с работающим браузером, например, на телефонах или Chromebook, даже при нестабильном подключении к Интернету.

Основные компоненты Brython

Ядро Brython содержится в *brython.js* и состоит из следующих компонентов:

- `brython()` – основная функция для выполнения скрипта `python` в теге `<script type="text/python">`, доступная в глобальном пространстве имен JavaScript. Она должна быть явно вызвана после загрузки DOM.
- `__BRYTHON__` – глобальный объект JavaScript, который содержит все внутренние объекты, необходимые для запуска скриптов Python. Этот объект не используется непосредственно при написании приложений Brython.
- `Built-in types` – реализации встроенных типов Python в JavaScript. Например, `py_int.js`, `py_string.js` и `py_dicts.js` являются соответствующими реализациями `int`, `str` и `dict`.
- `Browser` – модуль, который реализует в Python методы работы с DOM аналогичные стандартным методам JavaScript.

Стандартная библиотека Brython

Разработчики Brython компилируют стандартную библиотеку CPython в JavaScript и объединяют результат в пакет `brython_stdlib.js`.

Исходя из данной структуры, `brython` отлично подходит для запуска `python` скриптов на стороне клиента, предоставляя привычный для web-разработчика инструментарий, но реализованный на `python 3`. Так как Brython реализует удобный обмен данными между скриптом `python` и JavaScript, становится возможным взаимодействие путем GET/POST запросов с серверной частью любого приложения.

Однако у Brython есть свои ограничения, а именно отсутствие возможности включить в проект стороннюю библиотеку `python` не переписанную в `brython` на JavaScript.

На основе `brython` в итоге появилось дополнение к выше показанному интерактивному редактору кода в личных блогах студентов, который предоставляет студентам возможность писать и «налету» выполнять свой код `python` (рис. 3).

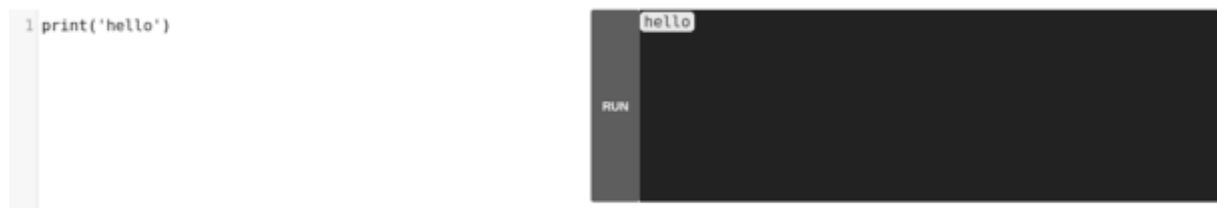


Рис. 7. Пример публикации кода `python` и результата его выполнения на персональном сайте студента [4]

Заключение. Интерактивный редактор кода, на личных блогах студентов, существенно содействует повышению эффективности учебно-образовательного процесса, особенно, при реализации дистанционных образовательных технологий:

1. Упростил выполнение практических и лабораторных работ студентами.

2. Облегчил процесс проверки этих работ преподавателем и дистанционные консультации по существу кода.

3. Дает возможность отказаться от поддержания большого инструментальных пакетов и сред на учебных и преподавательских компьютерах.

Показанные две версии интерактивного редактора поддерживают PHP, JavaScript, Python. Но данную методологию можно применить и к другим распространенным языкам программирования.

Библиографический список

1. Костерин, В.В. CMS, как технологическая основа непрерывной подготовки специалистов информационных технологий [Текст] / В.В.Костерин // Новые информационные технологии в образовании: материалы VI междунар. науч.- практ. конф. – Екатеринбург, 2013 – С. 354.
2. Костерин, В.В. Открытая рабочая среда поддержки учебного курса [Текст] / В.В.Костерин // Новые информационные технологии в образовании: материалы VII междунар. науч.- практ. конф. – Екатеринбург, 2014 – С. 74–78.
3. Официальный сайт Github – <https://github.com>
4. Бизнес-информатика – 2019 // Электронный учебно-методический журнал группы ЭУ-220 - <https://is20-2019.susu.ru/>
5. Официальный сайт Wordpress – <https://ru.wordpress.org>
6. Официальный сайт команды Brython – <https://brython.info>
7. Костерин, В.В. Brython: Python в вашем браузере [Текст] / В.В.Костерин // сайт «Курс молодого бойца: Python» – <https://chel-center.ru/python-yfc/2021/04/06/brython-python-v-vashem-brauzere/>
8. Официальный сайт Github – <https://github.com/>
9. Официальный сайт XAMPP – <https://www.xampp.ru/>
10. Официальная документация по PHP – <https://www.php.net/docs.php>
11. Клинышев, Д. 7 лучших интерпретаторов Python / Д. Клинышев // сайт DevGang – <https://dev-gang.ru/article/7-luczshih-perevodczikov-python-obnovleno-2019-auvtihzcfi/>