

*Академэнерго, Казанский научный центр,
Российская академия наук*

О свободных форматах публикации результатов научных исследований

Ф.С. Занько



1980-е

Microsoft, DOS/Windows, Word, Turbo Pascal...



2010-е

Google, Linux, OpenOffice, Python...

ВЫЖИВЕТ ЛИ КОММЕРЧЕСКОЕ ПО?



НОВЫЕ ТЕХНОЛОГИИ ПЕРЕДАЧИ ИНФОРМАЦИИ

Изобретение письма → государство,
литература, наука, университет...

Изобретение книгопечатания →
массовое школьное образование,
любительская наука

Интернет → свободное ПО,
свободное образование,
свободная наука



СИМПТОМЫ КРИЗИСА В НАУКЕ

- рост числа публикаций
- падение их среднего уровня
- «умирание» бумаги как основного носителя информации
- коммерческие издательства - препятствие на пути прогресса в науке

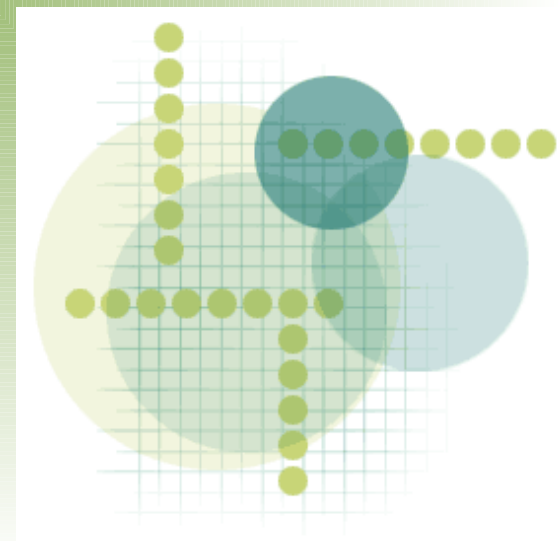


УЧЕНЫЕ ИЗ СЛАБОРАЗВИТЫХ СТРАН



1 статья - 30\$

1 год - десятки статей

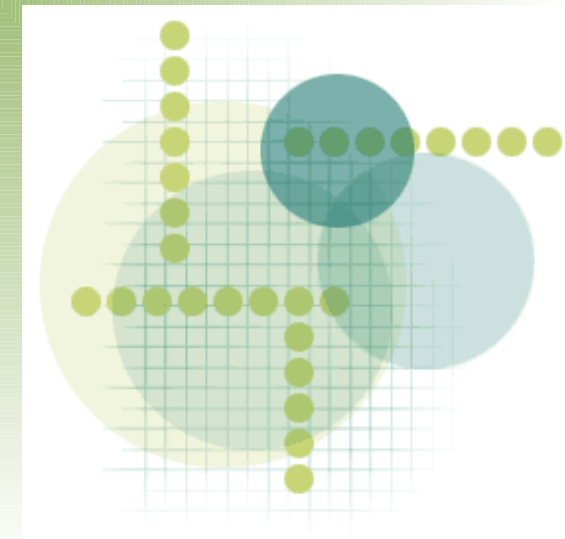


ГРИГОРИЙ ЯКОВЛЕВИЧ ПЕРЕЛЬМАН



- лауреат Филдсовской премии
- лауреат премии тысячелетия математического института Клэя

Импакт-фактор
(рейтинг публикаций) = 0

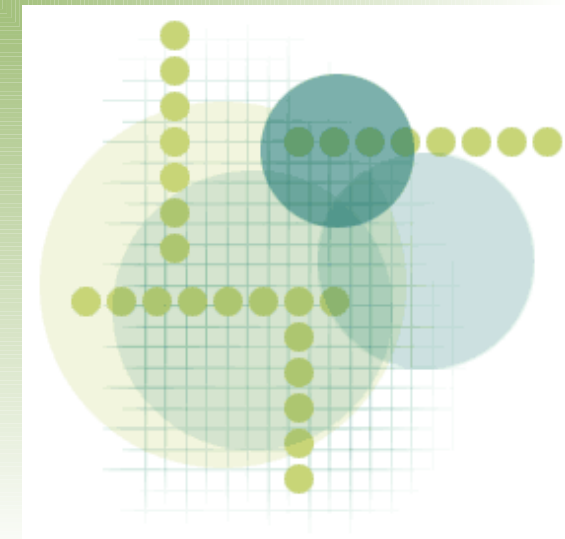


ФОРМАТЫ НАУЧНОЙ ДОКУМЕНТАЦИИ

<i>Формат</i>	<i>Преимущества</i>	<i>Недостатки</i>
Визуальные форматы WYSIWYG (OpenOffice и др.)	Простота освоения, широкое распространение, лёгкость получения печатного документа среднего качества	Нечитаемый код разметки, сложность работы с большими файлами, различные сбои; генерируемый HTML-код не читаем
TEX/PDF	Простой, читаемый, мощный язык математической разметки, отличное качество печатного документа; легко конвертируется в читаемый HTML-код; удобен для работы с большими файлами	Трудоёмкий в освоении и использовании из-за концентрации на печатном представлении документа
DocBook	Универсальный формат, легко конвертируется во все остальные основные форматы документации	Сложный и громоздкий, код плохо читаем
Облегчённые языки разметки (Markdown и т.п.)	Очень легко читается исходный текст, легко конвертируется во все остальные основные форматы	Облегчённый язык разметки — это посредник, ещё один язык, который нужно осваивать пользователю

ТРЕБОВАНИЯ К ФОРМАТАМ НАУЧНОЙ ДОКУМЕНТАЦИИ

- **открытый** и свободный
- основная форма документа - **электронная**, а не бумажная
- легко **читаемый** код разметки
- избегать языков- и форматов-**посредников**
- **время** важнее красоты



HTML 3.2 (без листов стилей)

- код разметки легко писать вручную
- код хорошо читается
- файлы легко использовать в сценариях
- исключительная переносимость
- сразу готов к публикации в Интернет

Примеры:

Предыстория термоанемометра. Рождение прибора

Хинсен К. Python в научной работе



КОММЕНТАРИИ PYTHON В СТИЛЕ HTML

Цель: публикация в Интернете текстов программ обработки научных данных так, чтобы они были понятны не только их автору

Реализация: программный код Python и код разметки HTML в «одном флаконе»

Описание:

[html2py.html](#)

[py2html.html](#)



ТЕКСТ ПРОГРАММЫ В БРАУЗЕРЕ

html2py.html - Mozilla Firefox


Файл Правка Вид Журнал Закладки Инструменты Справка

html2py.html 10-conference-Kaluga-2013.pdf (o... +

www.russianlutheran.org/python/literate/html2py.html григорий перельман

Python), а метке `code_start` - значение, равное 1 (в строке обнаружен стартовый тег `<CODE>`).

Затем в том же шаге цикла проводится аналогичный поиск тега `</CODE>`. Причем, если он обнаруживается, то не только меняются значения переменных `code` и `code_end`, но и перед самим этим тегом ставится значок `##`, превращающий оставшуюся часть строки в комментарий. Строка `line` разбивается на две части методом `split()`; границей раздела является комбинация символов `</CODE>`. После этого обязательно нужен оператор `break`, принудительно прекращающий поиск тегов в строке, так как строка из-за вставки удлиняется, и тот же самый тег будет находиться несколько раз.



Условие `if (start_tag==code_tag.code_start1()) | (start_tag==code_tag.code_start2()):` идентично условию `if (start_tag=='<CODE>') | (start_tag=='<code>')`: и `if (end_tag==code_tag.code_end1()) | (end_tag==code_tag.code_end2()):` идентично `if (end_tag=='</CODE>') | (end_tag=='</code>')`:. Здесь теги `<CODE>` и `</CODE>` - часть кода Python, и заменить угловые скобки на их символические обозначения (`<` и `>`) нельзя. Поэтому код Python, где явно присутствуют сочетания символов `<CODE>` и `</CODE>`, вынесен за пределы этого файла в отдельный модуль `code_tag.py`.

```
for line in b.readlines():
    code_start=0
    code_end=0

    for i in range(len(line)):
        if (i<(len(line)-5)):
            start_tag=line[i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]
            if (start_tag==code_tag.code_start1()) | (start_tag==code_tag.code_start2()):
                code=1
                code_start=1
        if (i<(len(line)-6)):
            end_tag=line[i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]+line[i+6]
            if (end_tag==code_tag.code_end1()) | (end_tag==code_tag.code_end2()):
                code=0
                code_end=1
            line=line[:i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]+line[i+6]
```

Kaluga2013 html2py.html GoldenDict zanko2013 zanko - Libre libreoffiKSnapshot KSnapshot ru 15:40

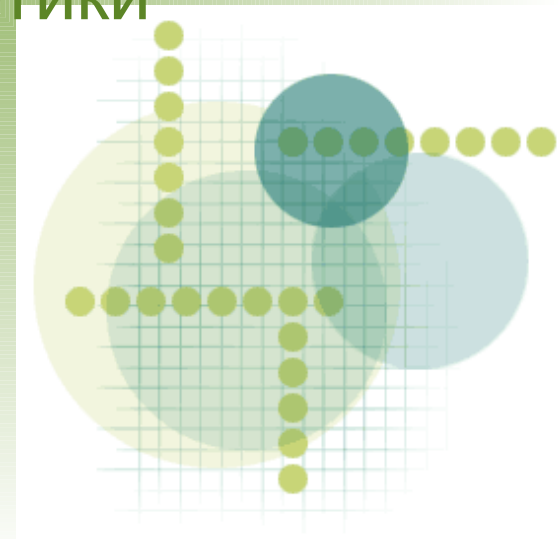
ТЕКСТ ПРОГРАММЫ В IDLE

```
html2py.py - /home/olimp_turb/Dropbox/Kaluga2013/html2py.py
File Edit Format Run Options Windows Help
##
##Затем в том же шаге цикла проводится аналогичный поиск тега <code>. Причем, если он обнаруживается, то не только
##
##<br>
##<br>
##Условие <code>if (start_tag==code_tag.code_start1()) | (start_tag==code_tag.code_start2()):</code> идентично условию <code>
##
##
##<pre><code>
    for line in b.readlines():
        code_start=0
        code_end=0

        for i in range(len(line)):
            if (i<(len(line)-5)):
                start_tag=line[i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]
                if (start_tag==code_tag.code_start1()) | (start_tag==code_tag.code_start2()):
                    code=1
                    code_start=i
            if (i<(len(line)-6)):
                end_tag=line[i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]+line[i+6]
                if (end_tag==code_tag.code_end1()) | (end_tag==code_tag.code_end2()):
                    code=0
                    code_end=i+6
                line_split=line.split(line[i]+line[i+1]+line[i+2]+line[i+3]+line[i+4]+line[i+5]+line[i+6])
                line=line_split[0]+'##'+code_tag.code_end1()+line_split[1]
                break##</code></pre>
##
##В зависимости от результатов поиска тегов <code> и </code>, рассматриваемая строка или приплюсовывается к ст
##
##<pre><code>
```

НЕДОСТАТКИ

- увеличивается трудоемкость программирования
- не получается сразу писать сценарий в виде HTML-файла
- исключается подсветка синтаксиса Python в браузере
- совмещение в одном файле двух языков не хорошо по соображениям стиля и эстетики
- и другие



СФЕРА ПРИМЕНЕНИЯ

- учебные, образцовые программы
- программы с серьезным математическим и/или научным содержанием



ВЫВОДЫ

- Если исследователь/преподаватель хочет, чтобы его труды читали, он должен не только создавать материалы высокого уровня, но и заботиться об их **свободном** распространении
- Рабочие форматы научной документации не должны создавать препятствий для ее свободного распространения

