Xonsh

Python 与 Shell 的碰撞

What is xonsh?

- Xonsh is a **Python-powered**, cross-platform, **Unix-gazing shell language** and **command prompt**.
- The language is a superset of Python 3.6+ with additional shell primitives that you are used to from Bash and IPython.
- It works on all major systems including Linux, OSX, and Windows. Xonsh is meant for the daily use of experts and novices.

它能做什么?

Show me the code!

```
import re
def runJudgeExe(cmd: list):
  pattern = re.compile(r"^得分: (\d+) \/ \d+$")
  output = $(@(judge) -i @(public_cases) -full_score 60 @(cmd))
  score = int(pattern.match(public_output.splitlines()[-1]).group(1))
  output = f"得分: {total_score}\n\n" + \
           f"评测输出: \n{output}\n\n"
  with open(os.path.join(cur_student_dir, "__judge_result__"), "w") as f:
    f.write(output)

def judgeCpp():
  mkdir -p build
  cd build
  cmake -DCMAKE_BUILD_TYPE=Release ..
  cmake --build .
  runJudgeExe(["./" + target_name])

def judgePython():
  runJudgeExe(["python", target_name + ".py"])
```

两种模式

- Python 模式
  - 可执行任意 Python 语句
  - 可引入一般的 Python 包
- Subprocess 模式
  - 如同一般的 Unix shell 一样，调用子进程
- 如何决定模式?
  - 例：`ls -l`

环境变量

- 用名字操作单个环境变量： `$HOME`
  - 可读，可写
  - 可以直接插值（仅限 subprocess 模式）

```
>>> echo "my home is $HOME"
my home is /home/snail
>>> echo @("my home is $HOME")
my home is $HOME
```

- 名字存储在 Python 表达式中： `${}`
  - `${'HO' + 'ME'}`
- 环境变量集合对象： `${...}`
- `CUDA_VISIBLE_DEVICES=0 python train.py`

```
with ${...}.swap(CUDA_VISIBLE_DEVICES='0'):
    python train.py
```

读取文件夹下的内容

- *：subprocess 模式下仍然可用
  - Python 模式下：g`*`
- Regular Expression Globbing
  - https://xon.sh/tutorial.html#regular-expression-globbing

```
>>> touch a aa aaa aba abba aab aabb abcba
>>> ls `a(a+|b+)a`
aaa  aba  abba
```

- 可以和既有的 Python 字面量机制，如 `f`，合用：

```
 mypattern = 'ab'
 print(f`{mypattern[0]}+`)
['a', 'aa', 'aaa']
```

在 Subprocess 命令行中使用 Python 变量

- for f in g`*`:
  - mv @(f) @(f.replace("txt", "json"))

Subprocess Types

- https://xon.sh/subproc_types.html
- Captured Subprocess with $() and !()
- Uncaptured Subprocess with $ and !
- @$
  - 语法糖，专用于解决 `$()` 输出带有 `\n` 的问题

其他你所熟悉的 shell 语法

- 它们都在那里！ `&&` `||` `>` `<` `>>` `|` …
- Job Control: 也和你熟悉的一样 `&` `fg` `bg`

其他功能

- 定义 Prompt: `$PROMPT` 变量
  - `$PROMPT = '{user}@{hostname}:{cwd} > '`
- 引用其他包/文件
  - Python/xonsh 均可
- Help?
- Aliases
- 代码补全

结构化 History

```
{'env': {...},  # Environment that xonsh was started with
 'sessionid': str, # UUID4 for the session
 'ts': [start, stop],  # start and stop timestamps for session [s since epoch]
 'locked': True,  # boolean for whether the file is in use or not
 'cmds': [  # array of commands
    {'inp': str,  # input command
     'ts': [start, stop],  # timestamps for the command
     'rtn': int, # command return code
     'out' str,  # stdout and stderr of command, for subproc commands
                 # this is only available select OSs. Off by default.
     },
    ...
    ],
}
```

- `history`
- `history show all`
- `history file`

Events

```
@events.on_chdir
def add_to_file(olddir, newdir, **kw):
    with open(g`~/.dirhist`[0], 'a') as dh:
        print(newdir, file=dh)
```

Programmable Tab-Completion

- `__xonsh__.completers`
    - `exclusive` vs `non-exclusive`

Bash to Xonsh Translation Guide

- https://xon.sh/bash_to_xsh.html