

传智播客

《Python 程序开发案例教程》

教学设计

课程名称：Python 程序开发案例教程

授课年级：2019 年级

授课学期：2019 学年第一学期

教师姓名：某某老师

2019 年 09 月 09 日

课题名称	第 11 章 正则表达式	计划学时	4 学时
内容分析	正则表达式是一种描述字符串结构的语法规则，在字符串的查找、匹配、替换等方面具有很强的能力，并且支持大多数编程语言，包括 Python。		
教学目标及基本要求	<ol style="list-style-type: none"> 1. 熟悉正则表达式的基础知识，包括字符和匹配规则 2. 掌握如何利用 re 模块实现预编译、匹配与搜索 3. 掌握使用 Match 对象中的方法 4. 掌握实现全文匹配的方法 5. 熟悉如何使用 re 模块实现检索替换、文本分割、贪婪匹配 		
教学重点	<ol style="list-style-type: none"> 1. 使用 match()函数进行匹配 2. 使用 search()函数进行匹配 3. 匹配对象 4. findall() 函数 		
教学难点	<ol style="list-style-type: none"> 1. 匹配对象 2. findall()函数 3. finditer()函数 4. 检索替换 5. 文本分割 6. 贪婪匹配 		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">（元字符、预定义字符集、re 模块、预编译、使用 match()函数进行匹配）</p> <p>一、创设情境，引出文件的打开与关闭操作</p> <p>（1）教师通过提出需求，引出什么是正则表达式。</p>		

正则表示“规则的”、“极好的”，正则表达式实际上就是规定了一组文本模式匹配规则的符号语言，一条正则表达式也称为一个模式，使用这些模式可以匹配指定文本中表达式模式相同的字符串。

(2) 明确学习目标

- 要求学生了解什么元字符
- 要求学生了解预定义字符集
- 要求学生了解 re 模块
- 要求学生掌握预编译
- 要求学生掌握使用 `match()` 函数进行匹配

二、进行重点知识的讲解

(1) 教师根据课件，讲述什么是元字符。

元字符指在正则表达式中具有特殊含义的专用字符，可以用来规定其前导字符在目标对象中出现的模式。正则表达式中的元字符一般由特殊字符和符号组成（具体详见教材表 11-1 说明）。

(2) 教师根据课件，讲述什么是预定义字符集。

正则表达式中预定义一些字符集，字符集能以简洁的方式表示一些由元字符和普通字符表示的匹配规则，例如 `\d` 可以匹配任意数字（具体详见教材表 11-2 预定义字符集）。

(3) 教师根据课件，讲述 re 模块。

Python 中的 re 模块是正则表达式模块，该模块提供了文本匹配查找、文本替换、文本分割等功能，例如 `compile()` 函数可以对正则表达式进行预编译，并返回一个 `Pattern` 对象（具体详见教材表 11-3 所示）。

(4) 教师根据课件，讲述预编译操作。

如果需要对一个正则表示重复使用，那么可以使用 `compile()` 函数对其进行预编译，以避免每次编译正则表达式的开销，其语法格式为：`compile(pattern, flags=0)`，参数 `pattern` 表示一个正则表达式，参数 `flags` 用于指定正则匹配模式。

(5) 教师根据课件，讲述使用 `match()` 函数进行匹配。

`match()` 函数检测目标文本的开始位置是否符合指定模式，若匹配成功返回一个匹配对象，否则返回 `None`，其语法格式为：`match(pattern, string, flags=0)`，参数

`pattern` 表示需要传入的正则表达式,参数 `string` 表示待匹配的目标文本,参数 `flags` 表示使用的匹配模式。

三、归纳总结，布置作业/随堂练习

(1) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括元字符、预定义字符集、`re` 模块、预编译、使用 `match()`函数进行匹配。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第二课时

(使用 `search()`函数进行匹配、实例 1：判断手机号所属运营商、匹配对象、`findall()`函数、`finditer()`函数)

一、回顾上节课的内容，继续讲解本课时的知识

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，继续介绍本课时的内容。

上节课我们学习了元字符、预定义字符集、`re` 模块、预编译、使用 `match()`函数进行匹配，本节课将带领大家学习使用 `search()`函数进行匹配、实例 1：判断手机号所属运营商、匹配对象、`findall()`函数、`finditer()`函数。

(3) 明确学习目标

- 要求学生掌握使用 `search()`函数进行匹配
- 要求学生了解实例 1：判断手机号所属运营商实现过程
- 要求学生掌握匹配对象
- 要求学生掌握 `findall()`函数的使用
- 要求学生掌握 `finditer()`函数的使用

二、进行重点知识的讲解

(1) 教师根据课件，讲解使用 `search()`函数进行匹配。

`search()`函数可以匹配出现在文本任意位置的字符串,如果匹配成功会返回一个匹配对象,否则返回 `None`,其语法格式为:`search(pattern,string,flags=0)`,其参数表示与 `match()`函数参数相同。

(2) 教师根据课件，讲解实例 1：判断手机号所属运营商的实现过程。

教师根据教学资源实现实例 1：判断手机号所属运营商，并能够向学生讲解实现过程。

(3) 教师根据课件，讲解匹配对象。

使用 `match()`函数和 `search()`函数进行正则匹配时，返回的不是单一的匹配结果，而是一组形如<_sre.SRE_Match object; span=(2, 4), match='ow'>的字符串，该组字符串表明返回结果是一个 `Match` 对象，其中主要包含两项内容，分别是 `span` 和 `match`，`span` 表示本次获取的匹配对象在原目标文本中所处的位置，目标文本的下标从 0 开始，`match` 表示匹配对象内容。

(4) 教师根据课件，讲解 `findall()`函数。

`findall()`函数可以获取目标文本中所有与正则表达式匹配的内容，并将所有匹配的内容以列表形式返回，`findall()`函数格式为：`findall(pattern,string,flags=0)`

(5) 教师根据课件，讲解 `finditer()`函数。

`finditer()`函数同样可以获取目标文本中所有与正则表达式匹配的内容，但该函数会将匹配到的字符串以迭代器的形式返回，`finditer()`函数格式为：`finditer(pattern,string,flags=0)`。

三、归纳总结，布置作业

(1) 回顾学习目标，总结本节课所学知识包括使用 `search()`函数进行匹配、

实例 1：判断手机号所属运营商、匹配对象、`findall()`函数、`finditer()`函数。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第三课时

(检索替换、实例 2：电影信息提取、文本分割、贪婪匹配、实例 3：用户注册验证)

一、回顾上节课内容，继续介绍本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。

(3) 明确学习目标

- 要求学生了解检索替换的使用
- 要求学生了解实例 2：电影信息提取实现过程
- 要求学生了解文本分割
- 要求学生了解贪婪匹配

二、进行重点知识的讲解

(1) 教师根据课件，讲解检索替换的使用。

`re` 模块中提供 `sub()`、`subn()` 函数用于替换目标文本中的匹配项，其格式为：
`sub/subn(pattern,repl,string,count=0,flags=0)`，参数 `pattern` 表示需要传入的正则表达式；参数 `repl` 表示用于替换的字符串；参数 `string` 表示待匹配的目标文本；参数 `count` 表示替换的次数；参数 `flags` 表示使用的匹配模式。

`sub()` 函数与 `subn()` 函数功能相同，不同的是若调用成功，`sub()` 函数会返回替换后的字符串，`subn()` 函数会返回包含替换结果和替换次数的元组。

(2) 教师根据课件，讲解实例 2：电信信息提取实现过程。

教师根据教学资源实现实例 2：电影信息提取，并能够向学生讲解实现过程。

(3) 教师根据课件，讲解文本分割的使用。

`re` 模块中提供的 `split()` 函数可使用正则表达式相同的字符串分割指定文本，其语法格式为：
`split(pattern,string,maxsplit=0,flags=0)`，参数 `pattern` 表示需要传入的正则表达式；参数 `string` 表示待匹配的目标文本；参数 `maxsplit` 表示用于指定分割的次数，默认值为 0，表示匹配指定模式并全部进行分隔，参数 `flags` 表示可选表示符。`split()` 函数调用成功后，分隔出的子项会被保存到列表中并返回。

(4) 教师根据课件，讲解什么是贪婪匹配。

正则表达式中有两种匹配方式：贪婪匹配和非贪婪匹配，所谓贪婪匹配，即在条件满足的情况下，尽量多的进行匹配，反之若尽量少地进行匹配，则为非贪婪匹配。

(5) 教师根据课件，讲解实例 3：用户注册验证实现过程。

教师根据教学资源实现实例 3：用户注册验证，并能够向学生讲解实现过程。

三、归纳总结，布置作业

(1) 回顾学习目标，总结本节课所学知识包括：检索替换、实例 2：电

	<p>影信息提取、文本分割、贪婪匹配、实例 3：用户注册验证。</p> <p>(2) 使用博学谷系统下发课后作业。</p> <p style="text-align: center;">第四课时 (上机练习)</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机一：（练习教材示例代码以及实例 1：判断手机号所属运营商、实例 2：电影信息提取、实例 3：用户注册验证）</p> <p>形式：独立完成</p> <p>要求：</p> <p>(1) 要求学生能够熟练掌握教材中示例代码。</p> <p>(2) 要求学生能够自己实现实例 1、实例 2 与实例 3 程序。</p>
<p>思考题 和习题</p>	<p>见教材第 11 章配套的习题</p>
<p>教 学 后 记</p>	