信息表达与智能处理

第1章 概述

第2章 万维网网页信息的表达及解析

第3章 Python语言入门与Web信息获取解析

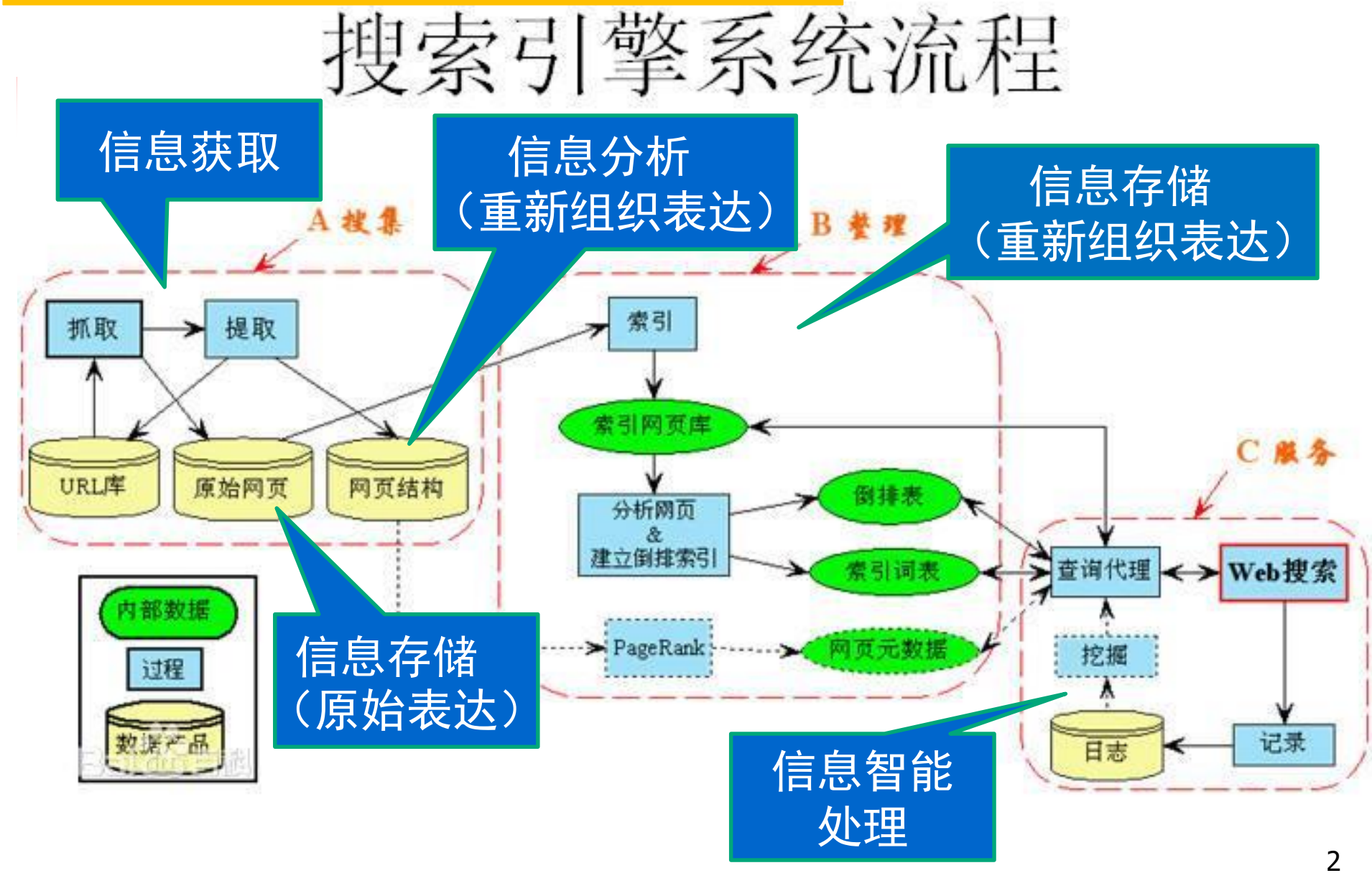
第4章 知识表示方法

第5章 知识图谱及资源描述框架RDF

第6章 正则表达式与Json、ASN.1

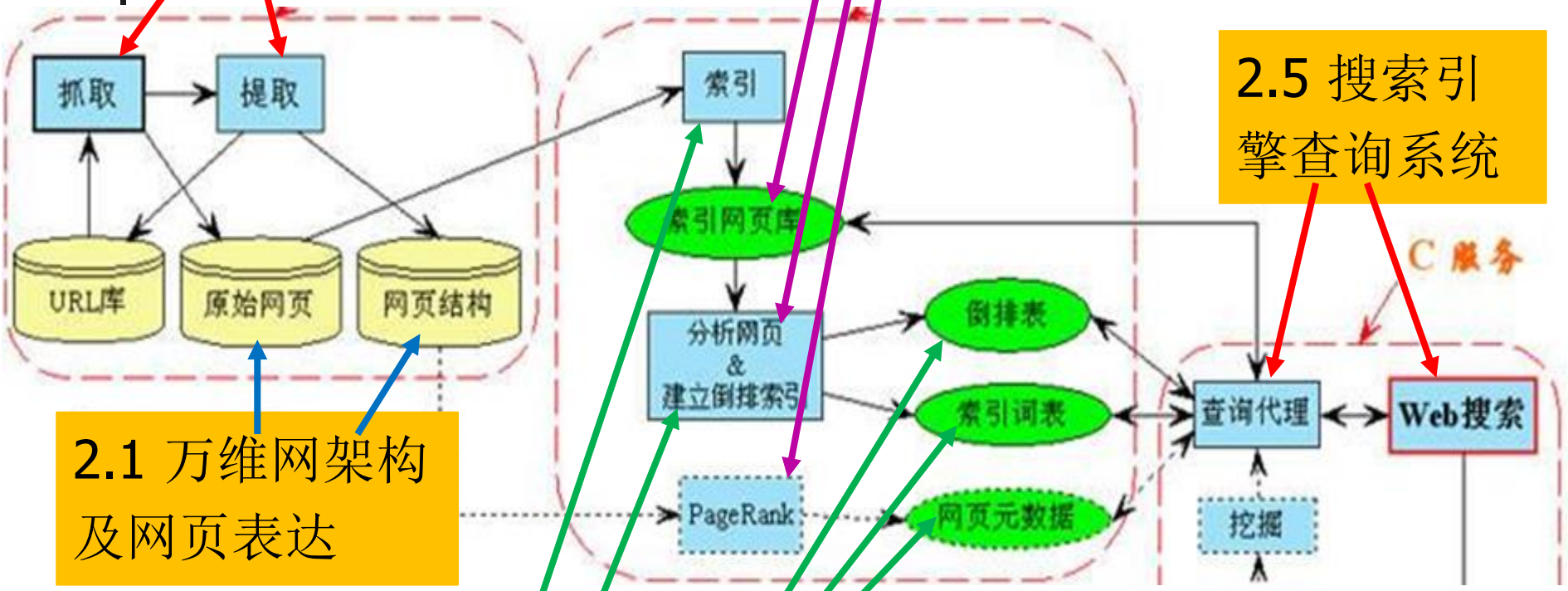
第7章 语义Web本体语言OWL

信息表达
信息智能处理的完整过程——



2.2 网页信息抽取

2.3 搜索引擎与分析系统



2.5 搜索引擎查询系统

2.1 万维网架构及网页表达

2.4 搜索引擎索引系统

第2章 万维网网页信息表达、获取及解析

2.1 万维网架构及网页表达

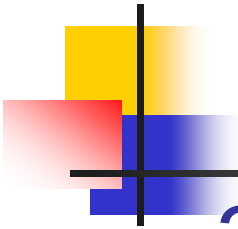
2.2 网页信息抽取

2.3 搜索引擎与分析系统

2.4 搜索引擎索引系统

2.5 搜索引擎查询系统5

《走进搜索引擎》 潘雪峰 花贵春 梁斌编著
电子工业出版社 2011年5月第2版



第2章 万维网网页信息的表达及解析

2.1 万维网架构及网页表达

2.1.1 HTML语言与半结构化网页

2.1.2 网页信息结构化

2.1.3 投票算法模型（获取完整正文）

2.1.1 HTML语言与半结构化网页

- 网页：
  - 网页是一个纯文本文件，它可以存放在世界某个角落的某一台计算机中，是万维网中的一“页”
  - 在WWW环境中，信息以页面信息组织，信息页面由语言来实现，在各个信息页面之间建立超文本链接以便浏览。
- **锚文本（anchor text）**
  - 网页中关于链接的一段描述，通常以文本和图片的方式出现
  - 可以指向文中的某个位置，也可以指向其他网页
  - HTML锚标签、锚文本：

```
<a href="http://zoujinsousuoyinq.com">走进搜索引擎</a>
```

2.1.1 HTML语言与半结构化网页

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>欢迎来到我的网站</title>
6  </head>
7  <body>
8      哇哦，这是我的第一个网页！
9  </body>
10 </html>
```

万维网上的网页数据具有一定的结构性，是由HTML标签带来的结构性，如：

- <TITLE>标签：标识网页主题
- <TD>标签：有些表示文章主题，有些表示文章段落，或者其他广告信息等。

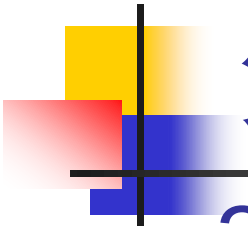
2.1.1 HTML语言与半结构化网页

- 结构化、半结构化、非结构化是按照数据格式分类

- **结构化数据**
  - 能够用统一的结构加以表示的信息
  - 业界指关系模型数据，即以关系数据库表形式管理的数据
- **非结构化数据**
  - 不符合任何预定义的模型，如WORD、PDF、PPT，各种格式的图片、视频等。
- **半结构化数据：**
  - 介于完全结构化数据和完全非结构化数据之间
  - 非关系模型的、有基本固定结构模式的数据，例如HTML文档、XML文档、JSON文档等

结构化数据：先有结构、再有数据
半结构化数据：先有数据，再有结构

数据模型：
结构化数据：二维表（关系型）
半结构化数据：树、图
非结构化数据：无



第2章 万维网网页信息的表达及解析

2.1万维网架构及网页表达

2.1.1 HTML语言与半结构化网页

2.1.2 网页信息结构化

2.1.3 投票算法模型

2.1.2 网页信息结构化

- HTML标签带来的结构性不能满足网页分析的需要
- 网页结构化是网页分析的首要任务
- **1.网页结构化的目标**
  - 针对搜索的需要，将半结构化的HTML网页中的数据按照如下几个基本属性依次抽取，生成一个网页对象。
    - 锚文本anchor
    - 标题title
    - 正文标题content title
    - 正文content
    - 正向链接link

2.1.2 网页

- 网页：“标题+锚文本” 描述
  - 例如清华大学主页可能被另外一些网页中存在锚（anchor）所指向，其锚文本就是该网页的最佳描述。
- 锚文本、标题和正文标题都是网页的简短描述

- 网页对象属性：

（1）锚文本：

- 对于某些没有标题的网页，锚文本是有益的补充

（2）网页标题：

- 描述网页的内容的属性
- HTML标识语言中<title></title>中间的文字部分，这部分文字表达了网页的基本含义。

（3）正文标题：

- 需要抽取正文中的适当文字作为正文标题。

2.1.2 网页信息结构化

- 网页对象（续）：

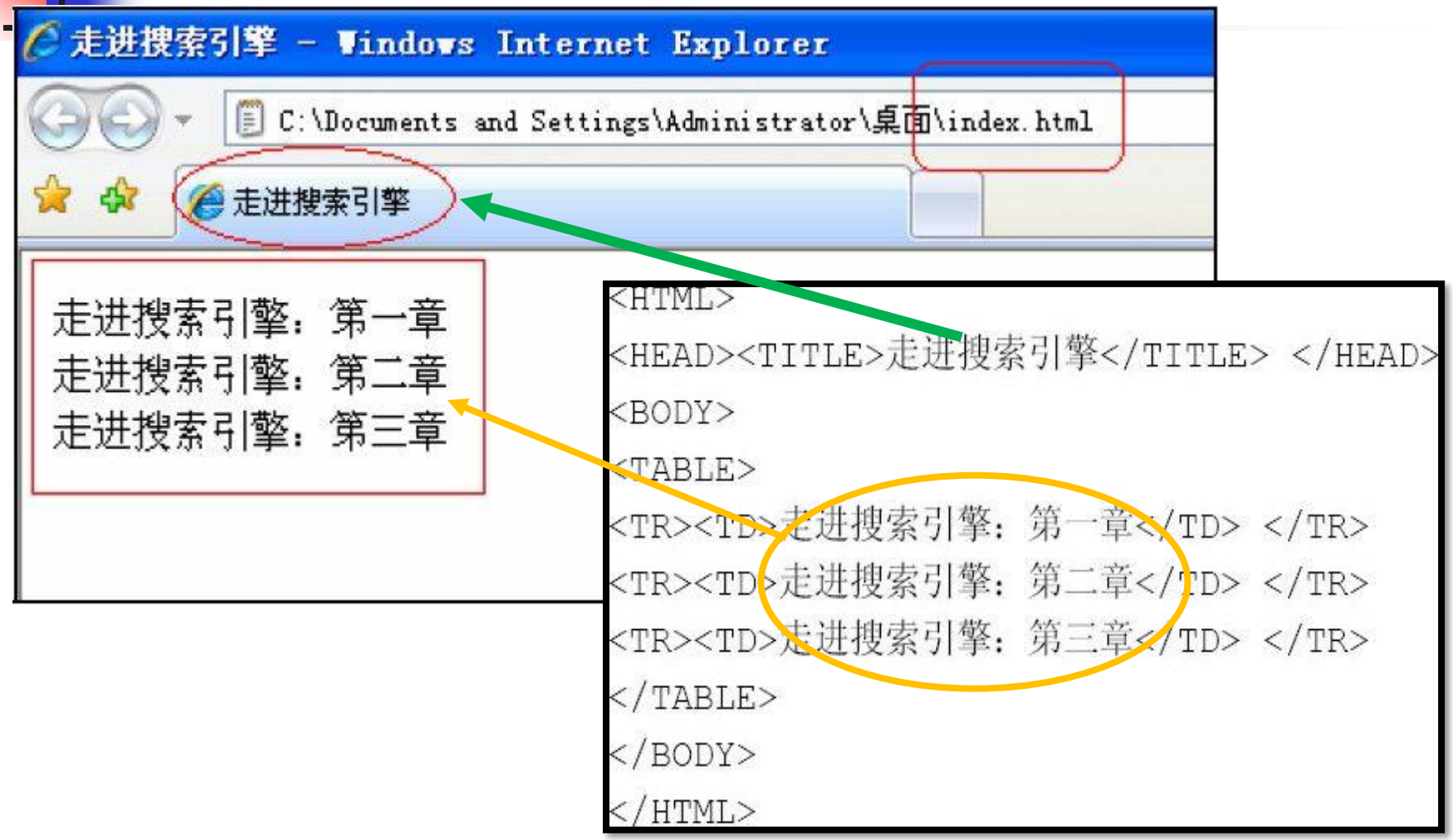
（4）正文：

- 是一个网页的主体内容，完整地表述了网页描述的信息
- 一般出现在<DIV>、<TABLE>、<CONTENT>和<P>等HTML标签中。

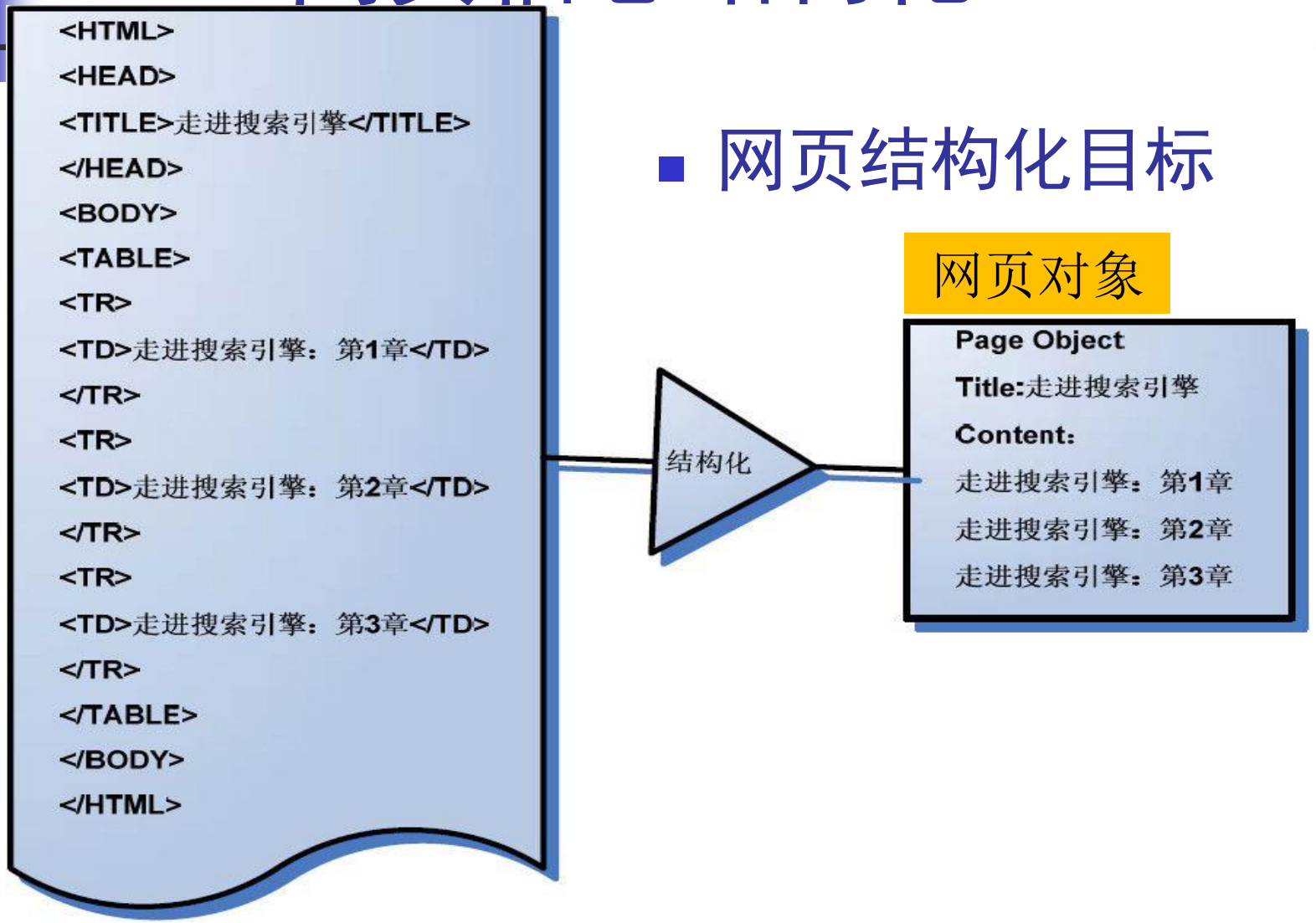
（5）正向链接：

- 网页制作者编写的引导用户继续在网上冲浪的链接，这些链接的文字也是其他网页的锚文本。

2.1.2 网页信息结构化


走进搜索引擎 - Windows Internet Explorer
C:\Documents and Settings\Administrator\桌面\index.html
走进搜索引擎
走进搜索引擎：第一章
走进搜索引擎：第二章
走进搜索引擎：第三章
<HTML>
<HEAD><TITLE>走进搜索引擎</TITLE> </HEAD>
<BODY>
<TABLE>
<TR><TD>走进搜索引擎：第一章</TD> </TR>
<TR><TD>走进搜索引擎：第二章</TD> </TR>
<TR><TD>走进搜索引擎：第三章</TD> </TR>
</TABLE>
</BODY>
</HTML>

2.1.2 网页信息结构化

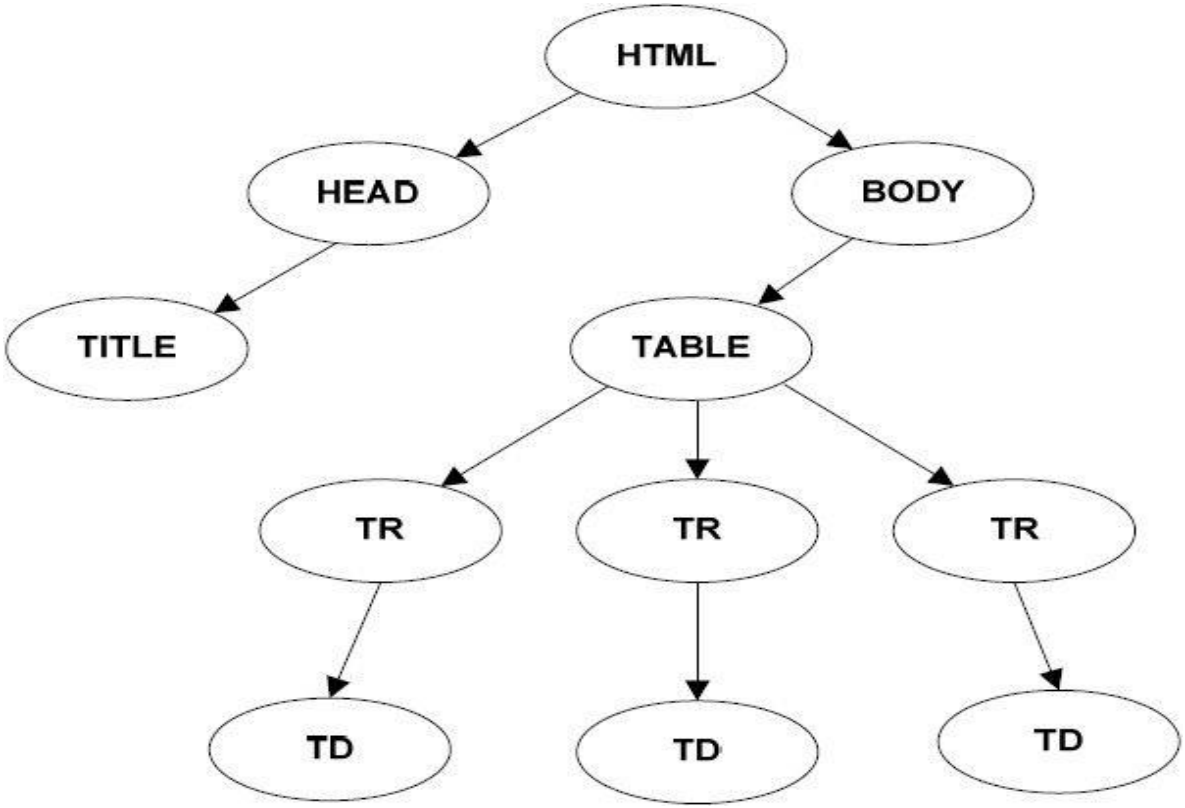


- 网页结构化目标

2.1.2 网页信息结构化

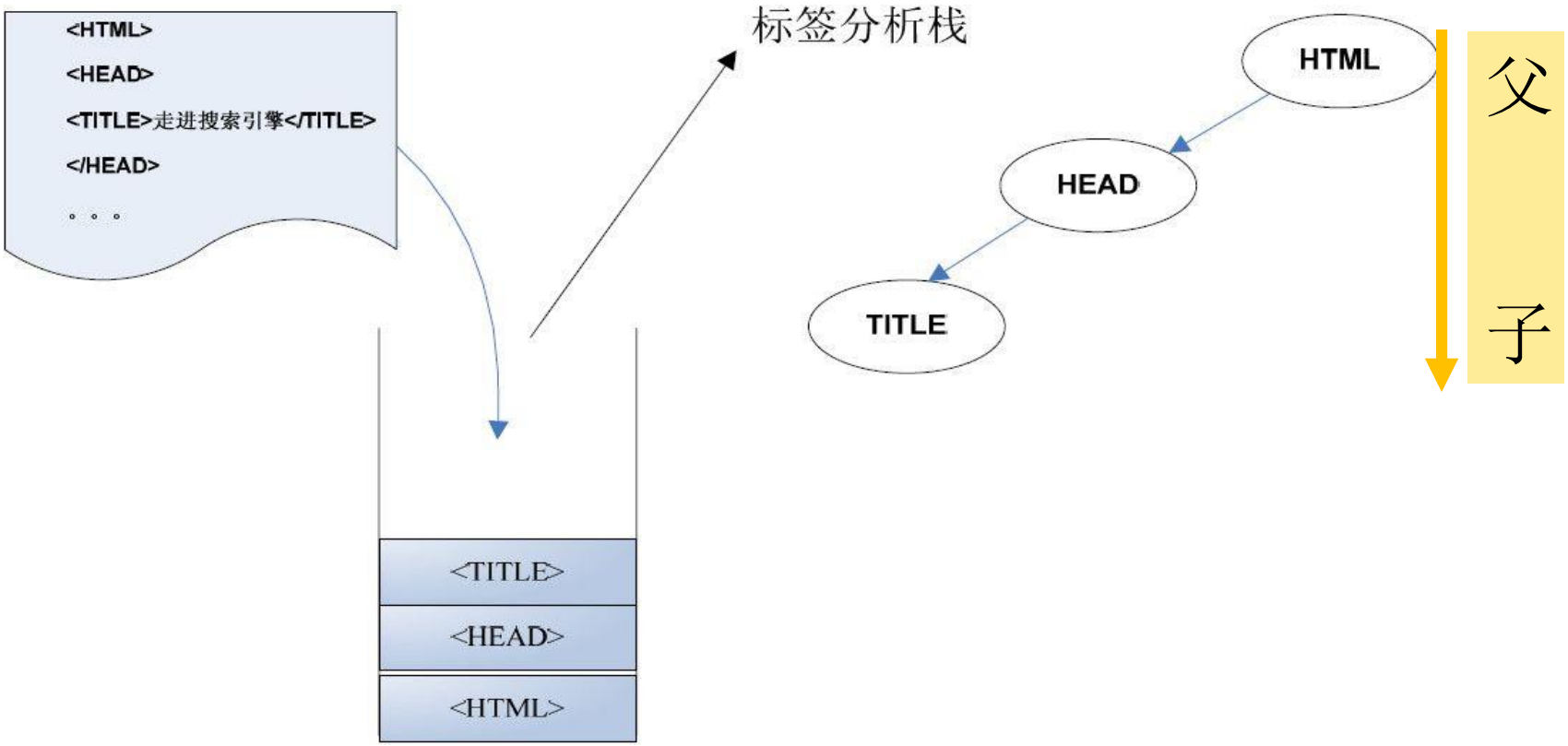
2. HTML标签树：

- 描述网页的HTML标签的嵌套关系（层次关系）

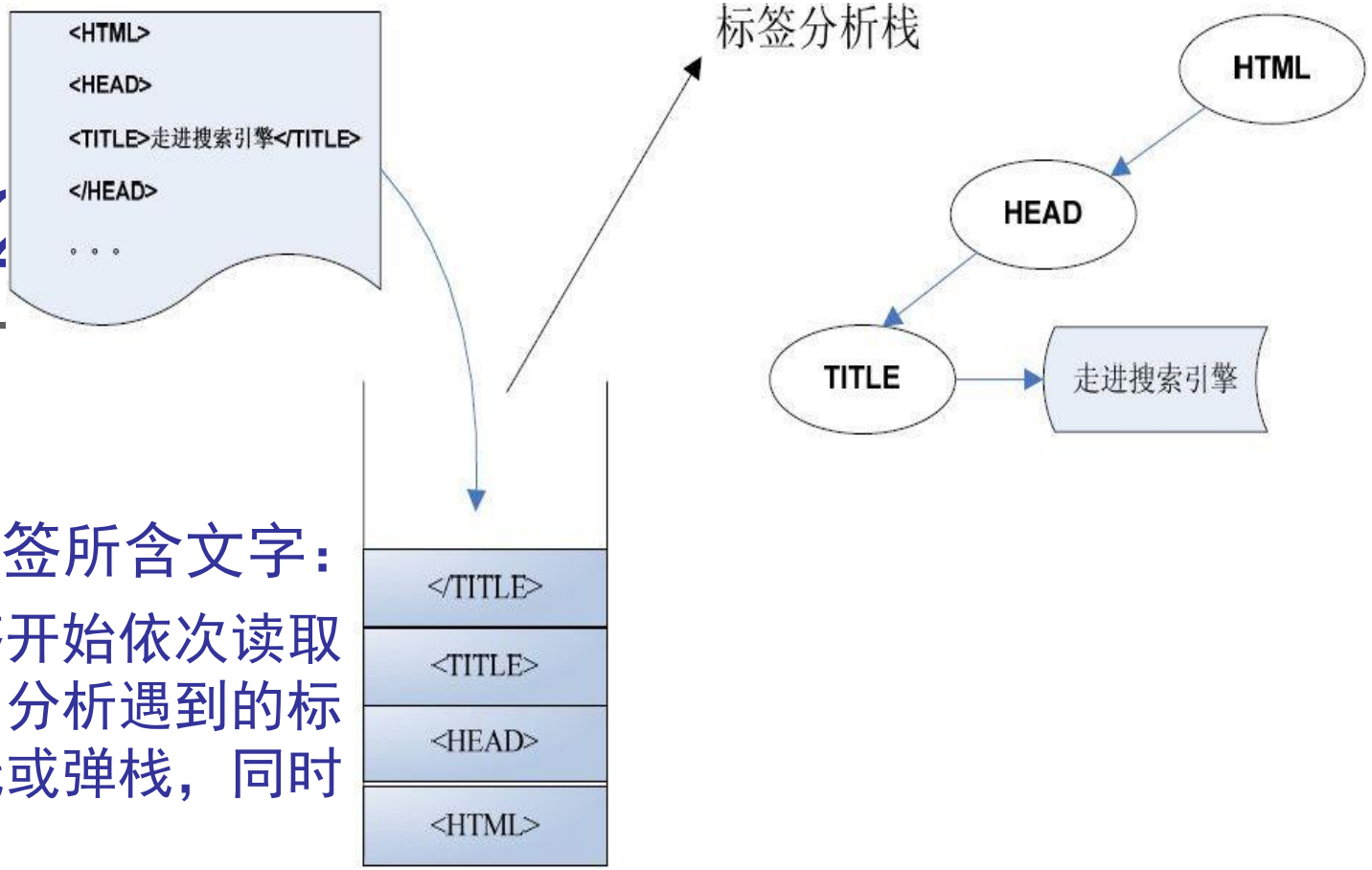


2.1.2 网页信息结构化

- 标签树建立过程
  - 1).标签分析栈：设一个站栈做标签分析的数据结构



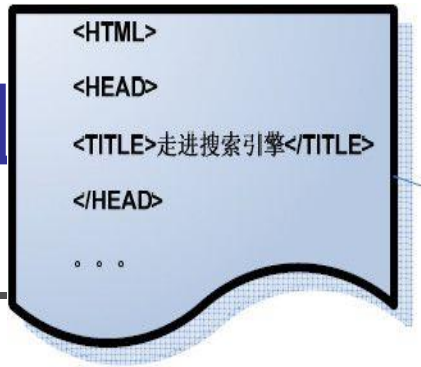
2.1.2

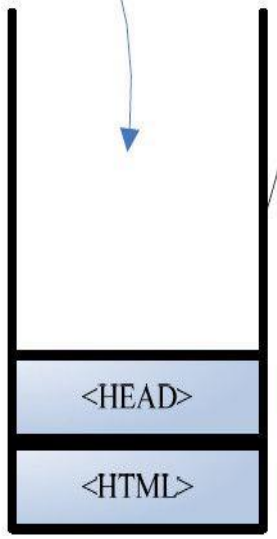


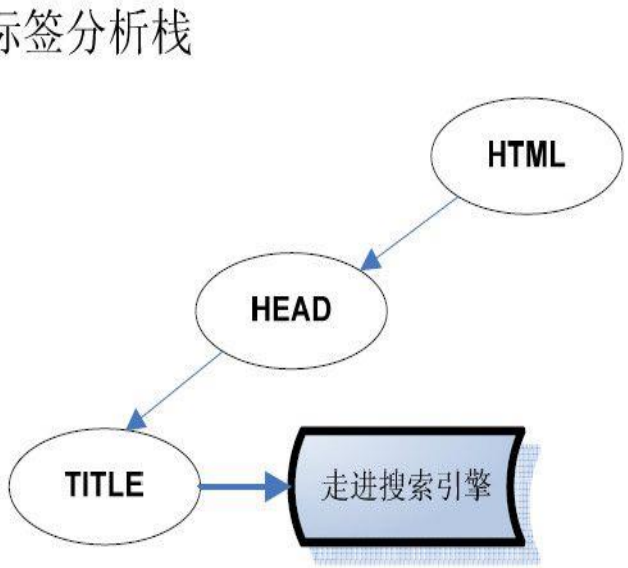
2).抽取TITLE标签所含文字：

- 程序从根标签开始依次读取网页源文件，分析遇到的标签，决定压栈或弹栈，同时建立标签树：
  - 根标签<HTML>压入栈底，并作为标签树根元素
  - 对起始标签，压栈，并按层次作为标签树元素
  - 对非标签，保留在树中不进栈，如果是正文标签的内容，则作为标签树相应标签的内容保留（本例“走进搜索引擎”）
  - （对结束标签，不压栈，判断如果与栈顶标签匹配，栈顶标签弹出）

2.1





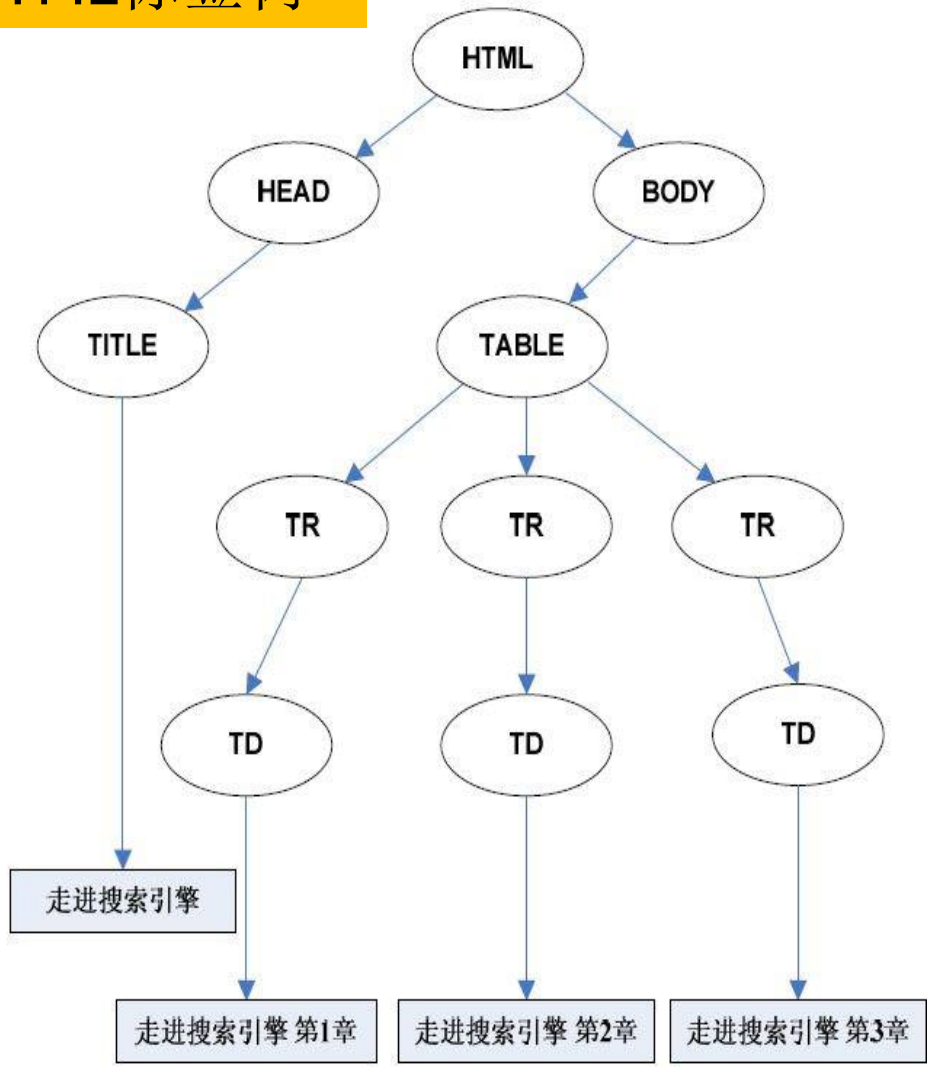


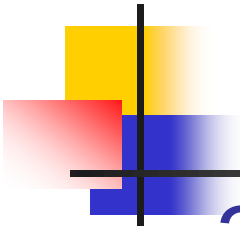
3).退栈过程

- 栈中相邻标签起、终成对，退栈
- 本例：
  - 标签栈顶的标签为<TITLE>，因此这两个成对标签同时退栈
  - 同时也确认了“走进搜索引擎”这一字符串为正文标题
  - 在标签树上的TITLE标签存放一个指向该字符串的指针，这样TITLE标签的文字被正确地抽取出来。

```
<HTML>
<HEAD>
<TITLE>走进搜索引擎</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD>走进搜索引擎：第1章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第2章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第3章</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

HTML标签树





第2章 万维网网页信息的表达及解析

2.1万维网架构及网页表达

2.1.1 HTML语言与半结构化网页

2.1.2 网页信息结构化

2.1.3 投票算法模型（获取完整正文）

2.1.3 投票算法模型

- 网页分析两个任务：
  - 获取网页标题
    - 通过HTML标签<TITLE>，容易获得
  - **获取网页完整正文**
    - 问题：
      - 网页中没有明显的标签标识出正文；
      - 正文可能分散在多个HTML标签中，如何组合出完整的正文
    - 工作
      - 获取文本块
      - 文本块组成完整正文

2.1.3 投票算法模型

- 文本块
  - 对于诸如<P></P>等标签间的文本
  - 如，“<TD>走进搜索引擎：第1章</TD>”
- 文本块3种类型
  - 主题型文本块（topic）
    - 大段文字的文本块，例:”<TD>走进搜索引擎：第1章</TD>”
  - 目录型文本块（hub）
    - 描述链接的文本块
    - 例:”走进搜索引擎：第1章”。
  - 图片型文本块（pic）
    - 描述图片的文本块
    - 例:”<img src="">走进搜索引擎：第1章</img>”。

可能包含广告等其他内容，必须与正文相区别

容易区分

2.1.3 投票算法模型

- “投票算法”判断哪个文本块是正文
- 正文抽取的投票算法的过程
  - 首先定义一系列规则
  - 通过这些规则为每一个文本块打分
  - 得分最高的被认为是正文的可能性足够大，并且可以接受
- 规则例（规则是要不断验证、调整的）

文本块文本的长度:

少于10个字，得分为0;

介于10～50个字得分为5分;

介于50～250个字，得分为8分;

超过250个字，得分为10分。

文本块文本位置:

在右侧，得分为0分;

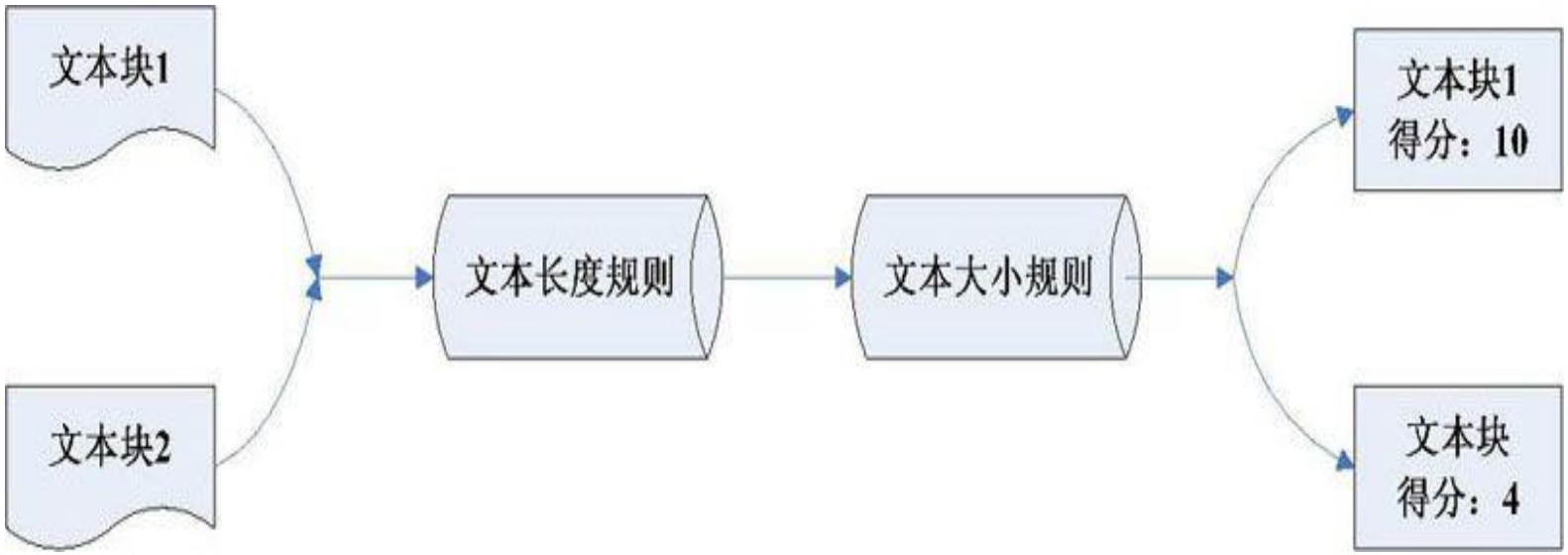
在顶部，得分为3分;

在左侧，得分为5分;

在中间，得分为10分

2.1.3 投票算法模型

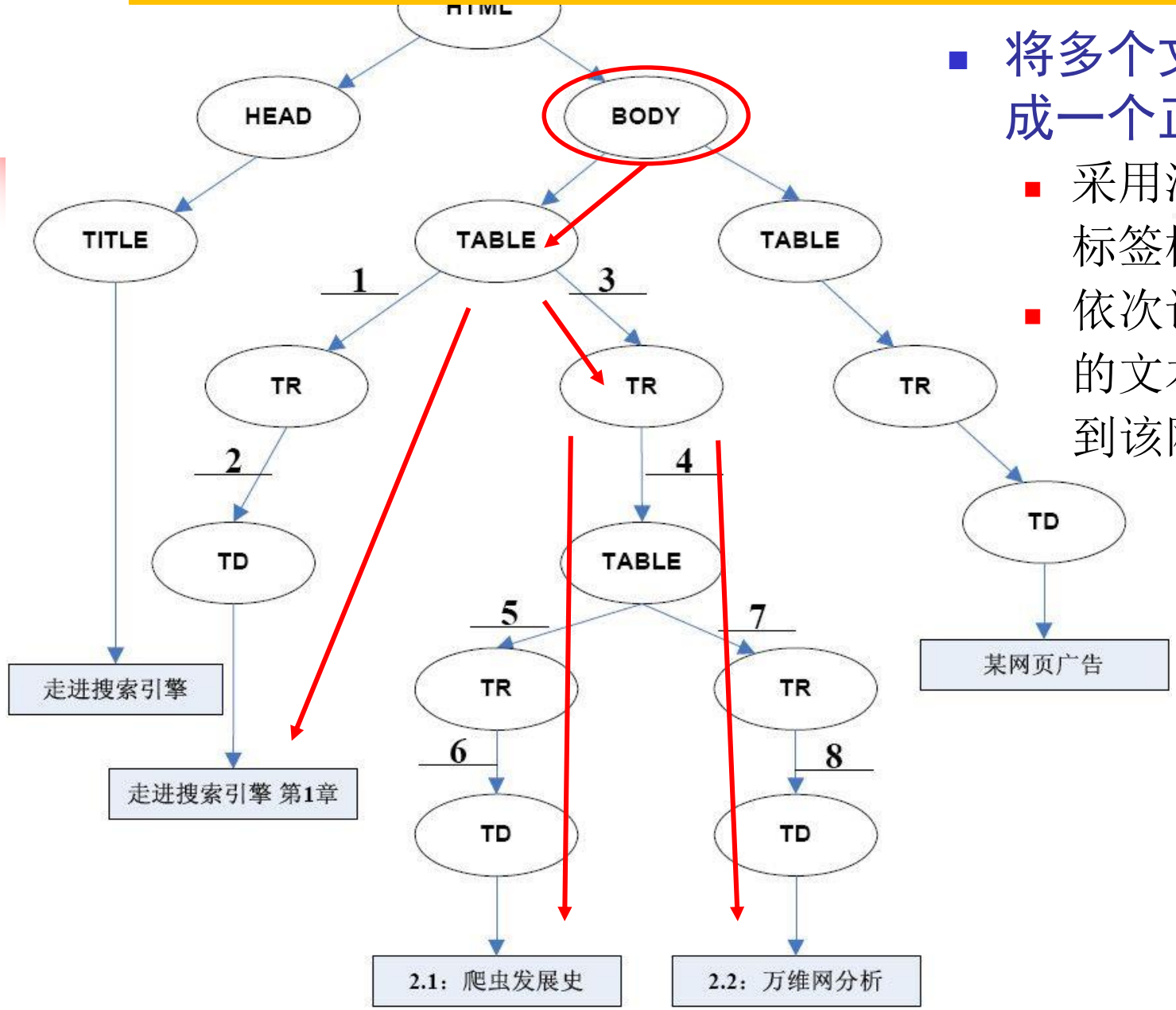
- 投票算法的过程
  - 依据不同的规则从不同的角度依次打分，文本块得分高的是正文的一部分



2.1.3 投票算法模型

- 规则器运用投票规则完成投票打分
- 规则器的打分是不断调整：
  - 规则数支持动态添加;
  - 规则打分计算过程：并行、串行
  - 规则的定义还需要通过足够多的网页进行反馈，之后才能得到一个公正客观的打分
  - 如果经常发现某些网页的正文段抽取错误，则要找出是哪一个规则器打分不合理才会导致这个结果
- 反复以上的过程，最后的打分将会趋于合理

走进搜索引擎第1章 2.1：爬虫发展史 2.2：万维网分析



- 将多个文本块组织成一个正文
  - 采用深度优先遍历标签树(后讲)
  - 依次记录主题类型的文本块，即可得到该网页的正文。

2.1.3 投票算法模型

- 图中的BODY标签下，第1个TABLE为正文，因为其下的文本块通过投票打分均为文本块
- 第2个TABLE标签下通过投票打分，其得分较低，因此可能为广告类信息
- 这样组织正文只需要从第1个TABLE开始深度优先遍历，遍历顺序为图中带下划线的数字所示
- 依次提取的文本块并按照顺序组织成如下的正文：

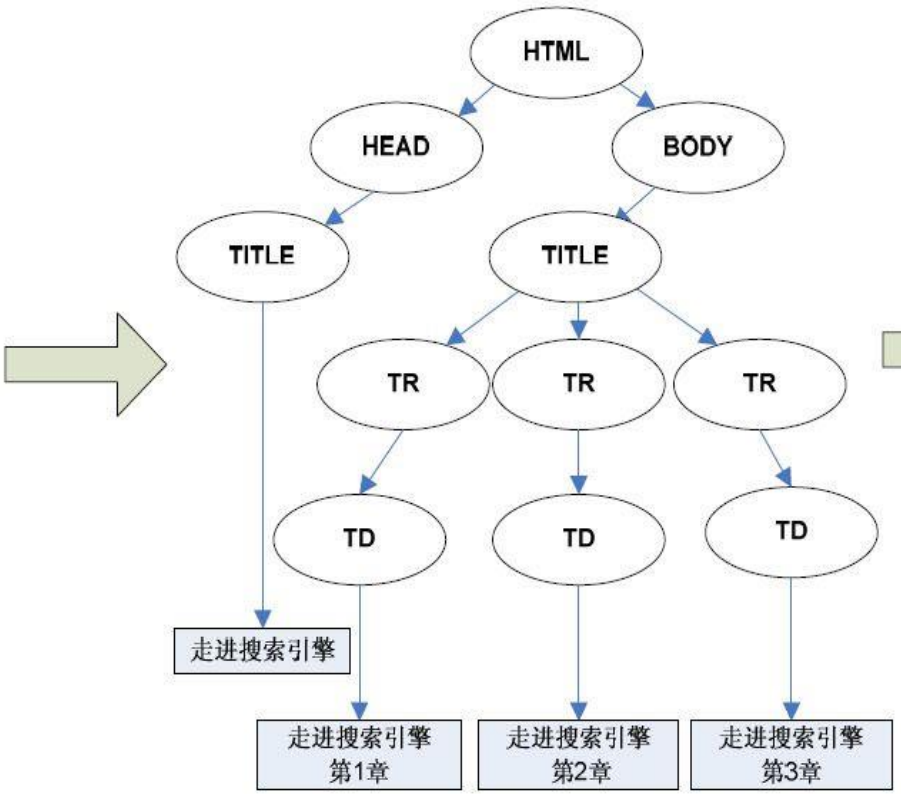
走进搜索引擎第1章 2.1：爬虫发展史 2.2：万维网分析

  - HTML标签的嵌套的特性，决定了深度优先遍历的顺序恰好能够组织成一个完整的正文。
- 对于其他的网页属性抽取，例如正文标题等也大多采用相同或类似的方法。

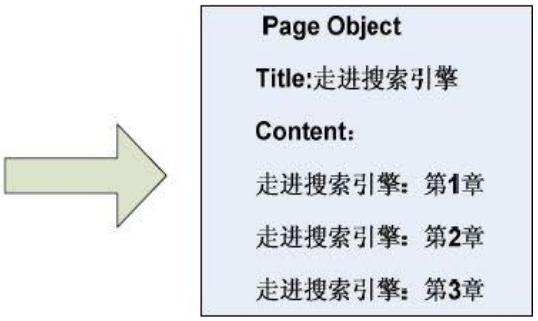
- 网页结构化完整过程：

```
<HTML>
<HEAD>
<TITLE>走进搜索引擎</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD>走进搜索引擎：第1章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第2章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第3章</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

1）建立标签树



2）投票获得正文



- 网页结构化意义：
  - 节约大量的存储
  - 保留网页有价值的信息，例如标题和正文；