

## 5. 웹 응용 2: 웹 크롤링

---

순천향대학교 컴퓨터공학과  
이 상 정

프로그래밍 기초 2

### 학습 내용

---

- 주피터 노트북
- HTML 문서
- 웹 크롤링과 뷰티풀 수프
- 기상청 페이지 웹 크롤링 예

## 주피터 노트북 (Jupyter Notebook)

- 파이썬의 IDLE 대신 웹 브라우저에서 파이썬 코드를 작성하고 실행하는 오픈 소스 웹 응용

- <https://jupyter.org/>

- 설치

- 윈도우 프롬프트 창에서 pip로 설치

**C:\W> pip install notebook**

```
C:\Windows\System32>pip install notebook
You are using pip version 6.0.8, however version 19.2.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Collecting notebook
  Downloading https://files.pythonhosted.org/packages/4e/b6/a6189ca7146482d93c912dbe6c65db0f264c1c88f707f707f707f707f707f707f/notebook-6.0.0-py3-none-any.whl (9.0MB)
    99% |#####| 9.0MB 20.5MB/s eta 0:00:01
```

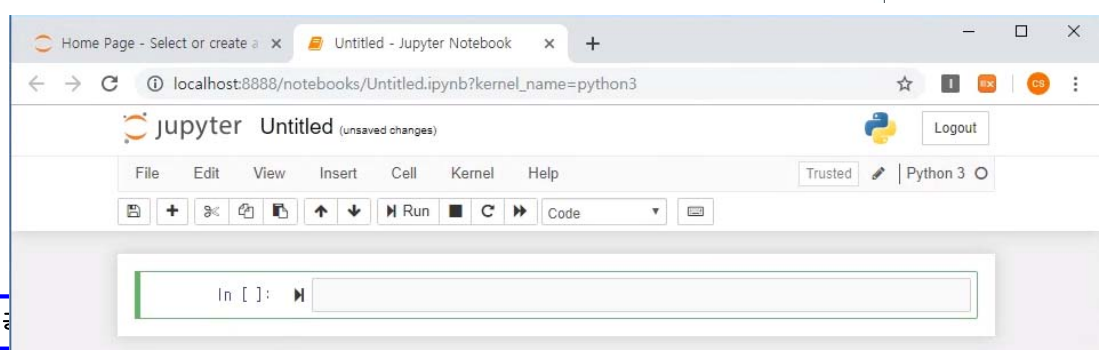
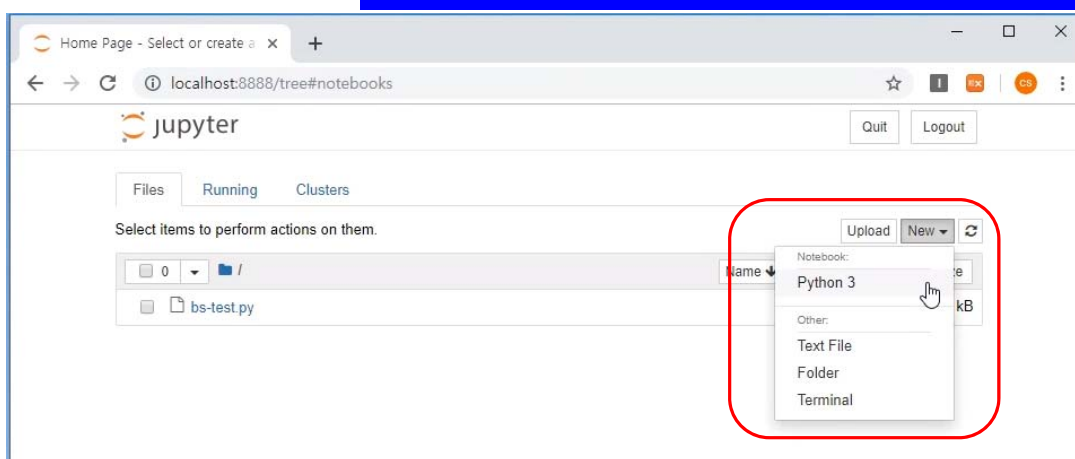
- 주피터 실행

**C:\W> jupyter notebook**

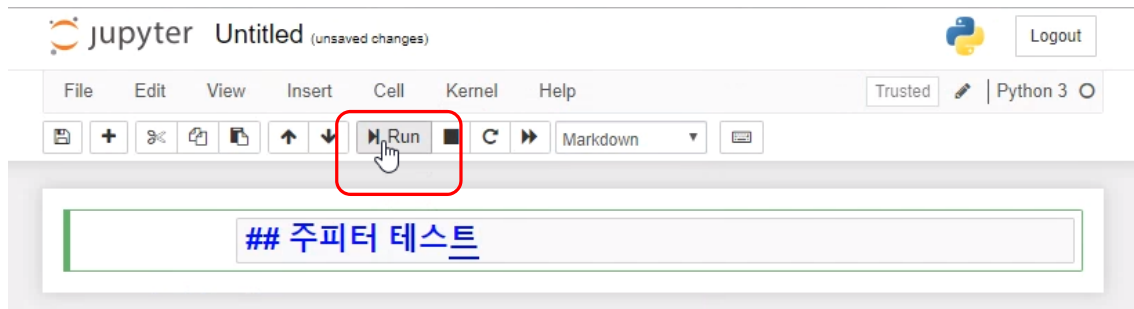
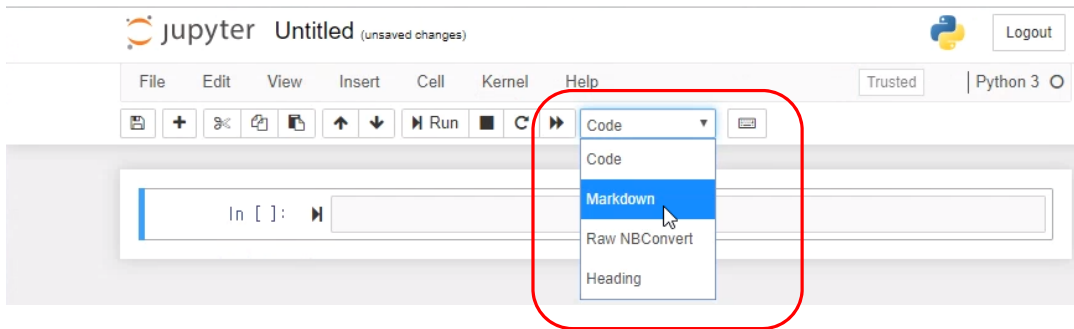
- 노트북 파일 저장 경로는 실행 시 아래와 같이 지정

**C:\W> jupyter notebook --notebook-dir 디렉터리**

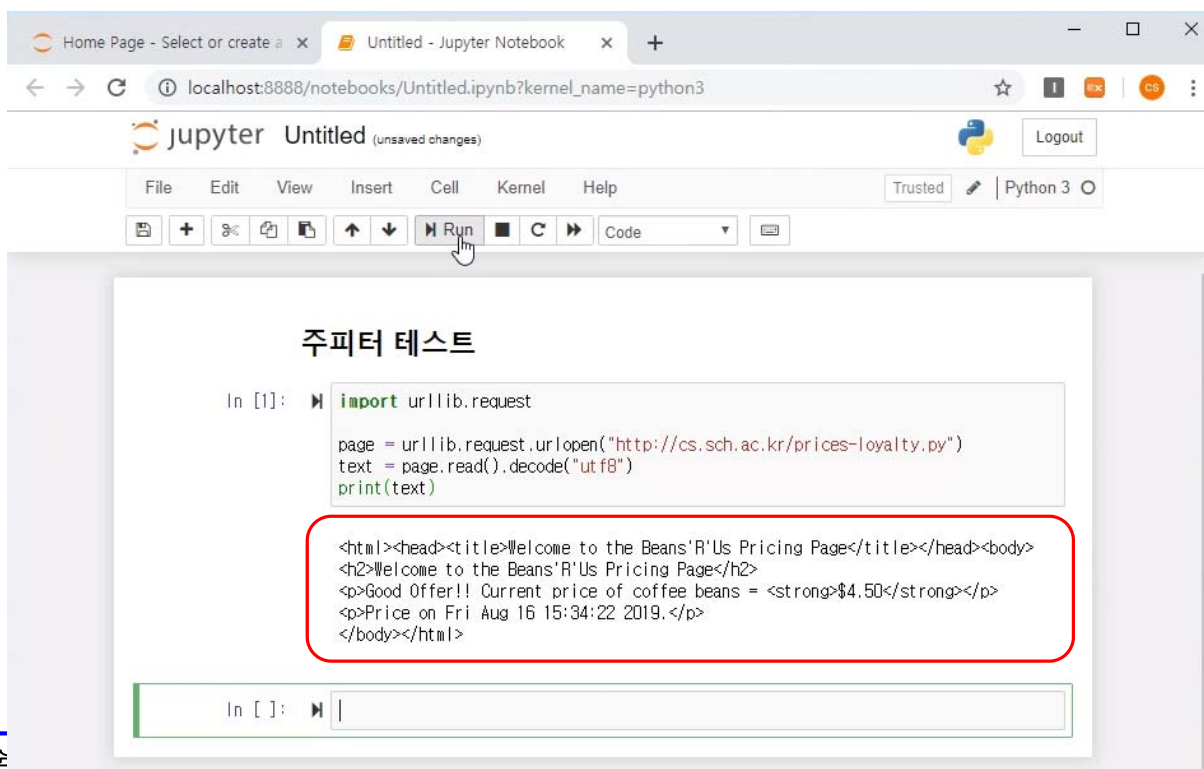
## 주피터 노트북 - 새 노트북 시작



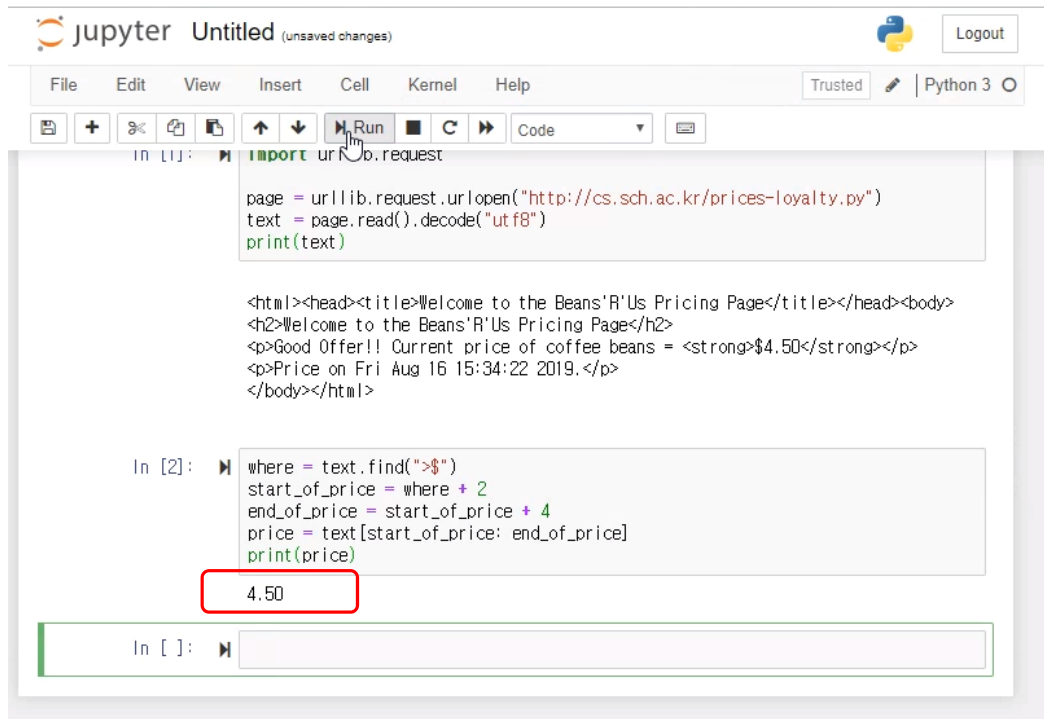
## 주피터 노트북 - 노트북 제목 삽입



## 주피터노트북 - 코드 작성 및 실행 (1)

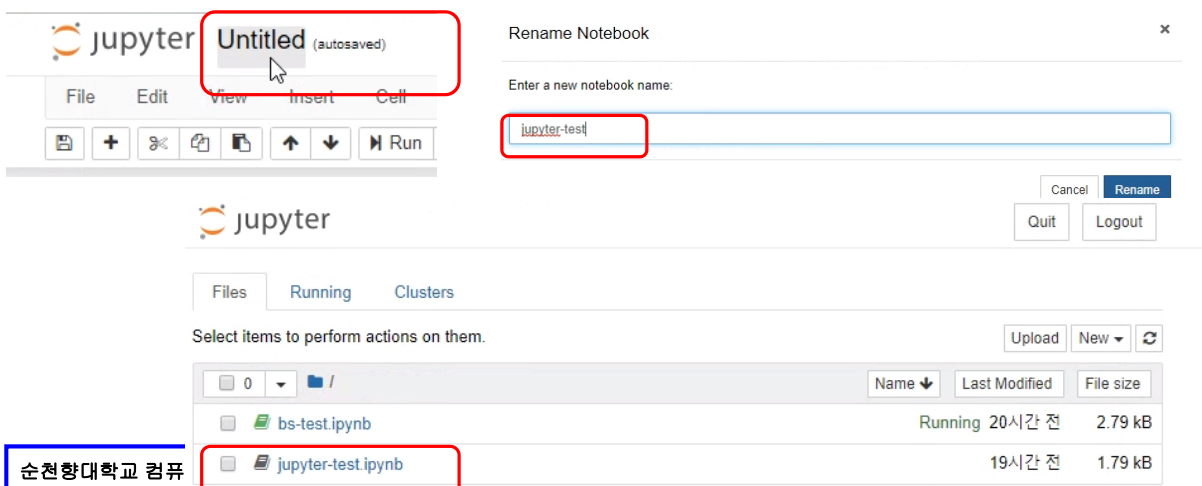


## 주피터노트북 - 코드 작성 및 실행 (2)



## 주피터 노트북 - 저장 및 이름 변경

- 노트북은 디폴트로 **Untitled.ipynb** 파일로 자동 저장
  - 주피터 노트북은 기존 파이썬 인터프리터에 각종 편의기능을 추가한 **Interactive Python** 사용
    - 파일 타입은 **.ipynb**
  - 파일 이름 변경 시 화면에서 **Untitled** 클릭하여 변경



## 주피터 노트북 - 파이썬 변환

- 주피터 노트북의 .ipynb 파일을 .py 파일로 변환 시 프롬프트 창에서 다음 명령 사용

- C:\W> jupyter nbconvert --to script [YOUR\_NOTEBOOK].ipynb
- jupyter-test.py 생성 예
  - jupyter nbconvert --to script jupyter-test.ipynb

```
D:\Wprog2>jupyter nbconvert --to script jupyter-test.ipynb
[NbConvertApp] Converting notebook jupyter-test.ipynb to script
[NbConvertApp] Writing 373 bytes to jupyter-test.py

D:\Wprog2>dir
D 드라이브의 볼륨: 코드_데이터
볼륨 일련 번호: F01A-1583

D:\Wprog2 디렉터리

2019-08-17 오전 10:44 <DIR>          .
2019-08-17 오전 10:44 <DIR>          ..
2019-08-17 오전 10:31 <DIR>          .ipynb_checkpoints
2019-08-16 오후 03:00          2,787 bs-test.ipynb
2019-08-16 오후 03:35          1,795 jupyter-test.ipynb
2019-08-17 오전 10:44          415 jupyter-test.py
                                4,997 바이트
                                3개 파일
                                3개 디렉터리 136,317,493,248 바이트 남음
```

## HTML 문서

- HTML (Hyper Text Markup Language)

- 웹에서 사용될 하이퍼 텍스트 문서를 만들기 위한 언어
- 태그(tag)를 기반으로 원하는 정보를 표현

```
<html>
  <head>
    <title>Welcome to the Beans'R'Us Pricing Page</title>
  </head>
  <body>
    <h2>Welcome to the Beans'R'Us Pricing Page</h2>
    <p>Good Offer!! Current price of coffee beans = <strong>$4.50</strong></p>
    <p>Price on Fri Aug 16 15:34:22 2019.</p>
  </body>
</html>
```

## HTML 기본 태그

시작	끝	설 명
<html>	</html>	HTML 문서의 시작과 끝
<head>	</head>	HTML 문서의 머릿글
<title>	</title>	HTML 문서의 제목
<body>	</body>	HTML 문서의 본문
<h2>	</h2>	제목을 나타내는 헤딩(heading) 태그 <h1>에서 부터 <h6>까지 있으며 작은 숫자일 수록 글자 크기가 큼
<p>	</p>	문단(paragraph)을 나타내는 태그
<strong>	</strong>	글자를 진하게 표시, <b> 태그와 같음
<!--	-->	주석(코멘트)
<a href=...>	</a>	하이퍼링크 태그 (anchor)
<table>	</table>	테이블의 시작과 끝을 표시
<th>	</th>	테이블의 제목 (table head)
<tr>	</tr>	테이블의 행 (table row)
<td>	</td>	테이블의 데이터 (table data)

## 웹 크롤링과 뷰티풀 수프 (Web Crawling and BeautifulSoup)

- 웹 크롤링은 웹 페이지에서 원하는 정보를 추출하는 소프트웨어 기법
  - 웹 스크레이핑(Web Scraping)이라고도 함
  - 웹 크롤러(web crawler)는 페이지를 방문해서 자료를 수집하는 일을 하는 프로그램
- 파이썬에서는 웹 크롤링을 위한 뷰티풀 수프(Beautiful Soup) 라이브러리를 제공
  - HTML, XML 문서를 구문 분석(파싱, parsing)하는 라이브러리
  - <https://www.crummy.com/software/BeautifulSoup/>
  - 라이브러리 설치  
C:W> pip install beautifulsoup4

## □ 뷰티풀 수프 객체 생성

- `from bs4 import BeautifulSoup`  
`soup = BeautifulSoup(html, 'html.parser')`
  - `html`: 읽어들이는 HTML 문서
    - `urllib.request.urlopen()` 등으로 읽어들이는 HTML 문서

## □ HTML 문서 태그 검색 메서드

- `soup.find_all(태그이름, 태그 속성)`
  - 지정된 태그 모두를 검색, 검색 결과는 리스트
  - 태그 속성은 {'속성명': '값', ...} 으로 기술
- `soup.find(name, attrs)`
  - 지정된 첫 번째 태그만 검색

# HTML 문서 읽기

`import urllib.request``page = urllib.request.urlopen("http://cs.sch.ac.kr/prices.py").read()``print(page)` # 바이트 문자열`print(page.decode("utf8"))` # 유니코드 문자열

# 뷰티풀 수프 객체 생성

`from bs4 import BeautifulSoup``soup = BeautifulSoup(page, 'html.parser')`

# 문단 태그 모두 검색

`p_all = soup.find_all("p")``print(p_all)`

# 검색한 문단 리스트 출력

`print(p_all[1])`

# 리스트의 두번째 항목만 출력

`print(p_all[1].text)`

# 태그를 제외한 내용만 출력

`print("-----")``for p in p_all:`

# 리스트 각 항목 내용 출력

`print(p.text)`

# 첫 강조 태그 검색

`s = soup.find("strong")``print(s)``print(s.text)`

# 태그를 제외한 내용만 출력

## 뷰티풀 수프 - 사용 예

## 뷰티풀 수프 - 실행 예 (1)

## Beautiful Soup 테스트

```
In [1]: # HTML 문서 읽기
import urllib.request

page = urllib.request.urlopen("http://cs.sch.ac.kr/prices.py").read()
print(page) # 바이트 문자열
print(page.decode("utf8")) # 유니코드 문자열
```

```
b'<html><head><title>Welcome to the Beans'R'Us Pricing Page</title></head><body>
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
<p>Current price of coffee beans = <strong>$6.59</strong></p>
<p>Price on Sat Aug 17 11:31:51 2019.</p>
</body></html>'
```

```
<html><head><title>Welcome to the Beans'R'Us Pricing Page</title></head><body>
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
<p>Current price of coffee beans = <strong>$6.59</strong></p>
<p>Price on Sat Aug 17 11:31:51 2019.</p>
</body></html>
```

```
In [4]: # 뷰티풀 수프 객체 생성
from bs4 import BeautifulSoup
soup = BeautifulSoup(page, 'html.parser')
```

## 뷰티풀 수프 - 실행 예 (2)

```
In [5]: # 문단 태그 모두 검색
p_all = soup.find_all("p")
print(p_all) # 검색한 문단 리스트 출력
print(p_all[1]) # 리스트의 두번째 항목만 출력
print(p_all[1].text) # 태그를 제외한 내용만 출력
print("-----")
for p in p_all: # 리스트 각 항목 내용 출력
    print(p.text)
```

```
[<p>Current price of coffee beans = <strong>$6.59</strong></p>, <p>Price on Sat
Aug 17 11:31:51 2019.</p>]
<p>Price on Sat Aug 17 11:31:51 2019.</p>
Price on Sat Aug 17 11:31:51 2019.
-----
Current price of coffee beans = $6.59
Price on Sat Aug 17 11:31:51 2019.
```

```
In [6]: # 첫 강조 태그 검색
s = soup.find("strong")
print(s)
print(s.text) # 태그를 제외한 내용만 출력
```

**\$6.59**  
\$6.59



# 기상청 페이지 웹크롤링 예

## 기상청의 현재 날씨 관측자료를 크롤링하여 파일로 저장하는 예

- 기상청의 관측자료에서 도시별 현재 날씨 페이지

- <http://www.weather.go.kr/weather/observation/currentweather.jsp>

The screenshot shows the '기상청 날씨누리' (KMA Weather) website. The '관측자료' (Observation Data) tab is selected. The page displays a table of current weather data for various cities. The table has columns for location, current weather, temperature, humidity, wind speed, and pressure. The data is for the date 2019.08.17 at 11:00.

지역	현재날씨	기온(℃)	습도(%)	풍속(km/h)	기압(hPa)
서울	구름없음	27.4	66	서북서 7.2	1000.7
부산	구름조금	25.3	75	서남서 18.4	999.3
인천	흐림	26.0	72	서 7.9	1001.1
수원	약한비단속	26.8	73	서남서 11.2	1001.2
대구		25.6	78	서남서 7.9	1001.4

순천향대학교 컴퓨터

5. 웹 응용 2

# 기상청 페이지 분석

## 웹 크롤링을 위해서는 해당 페이지의 화면과 HTML과의 연관성을 분석해야 함

- 크롬(Chrome) 웹 브라우저의 개발자 도구 활용
- F12 키 누르면 개발자 도구 표시

The screenshot shows the same website as before, but with the Chrome Developer Tools open. The 'Elements' panel is visible on the right, showing the HTML structure of the page. The 'body' element is selected, and its class is 'body-wrapper'. The 'body' element is highlighted in the DOM tree.

순천향대

응용 2

## 기상청 페이지 분석 - 테이블 데이터

### 기상청 페이지 중에서 테이블에 표시된 데이터를 검색

기상실황표 2019.08.17.12:00

지점	날씨				기온(℃)			강수		바람		기압(hPa)
	현재일기	시정 km	운량 1/10	중하운량	현재 기온	이슬점 온도	볼래 지수	일강수 mm	습도 %	풍향	풍속 km/h	해면 기압
서울	비 끝남	14.0	9	9	27.1	21.6	77	0.0	72	서	10.4	1000.5
백령도	구름조금	20 이상	3	3	25.1	19.9	74		73	서남서	19.8	999.2
인천	흐림	20 이상	9	9	26.3	20.8	76	0.0	72	서남서	11.2	1000.8
수원	약한비단속	14.0	9	9	26.2	22.4	77	6.6	80	북서	6.5	1001.1
등두천		18.1			26.4	22.0	77		77	서	8.3	1001.0
파주		20 이상			27.7	21.4	78		69	서	9.4	1000.9
강화		20 이상			27.5	19.8	77		63	남서	14.8	1001.0
양평		10.5			25.3	22.5	76	4.0	85	남동	3.2	1001.1
이천		17.3			27.7	21.7	78		70	남남서	20.9	1000.9
북춘천	구름많음	20 이상	8	8	27.9	20.9	78		66	남남서	11.5	1000.1
북강릉	구름많음	19.6	8	8	27.7	18.1	76		56	남서	11.2	999.0
울릉도	맑음	20 이상	0	0	28.0	20.5	78		64	남서	20.9	1001.2
속초		20 이상			30.9	18.6	79		48	서	7.9	998.7
철원		20 이상			26.6	22.4	77	0.7	78	남서	8.6	1000.6

순천향대학교 컴퓨터공학과

19

5. 웹 응용 2

## HTML 테이블

번호	이름
1	김영희
2	김순자

```

<html>
<head>
</head>
<body>
<table border="1">
  <tr>
    <th>번호</th>
    <th>이름</th>
  </tr>
  <tr>
    <td>1</td>
    <td>김영희</td>
  </tr>
  <tr>
    <td>2</td>
    <td>김순자</td>
  </tr>
</table>
</body>
</html>

```

<table>	</table>	테이블의 시작과 끝을 표시
<th>	</th>	테이블의 제목 (table head)
<tr>	</tr>	테이블의 행 (table row)
<td>	</td>	테이블의 데이터 (table data)

속성	비고
border	테이블의 테두리
bordercolor	테이블의 테두리 색상
width	테이블 가로 크기
height	테이블 세로 크기
align	정렬
bgcolor	배경색
colspan	가로 합병(열 합병)
rowspan	세로 합병(행 합병)

# 기상청 페이지 분석 - 테이블 시작 태그 (1)

- 개발자 도구에서 페이지 요소를 선택하는 탭을 누른 후 테이블의 첫 칸 지정하여 테이블 관련 부분 HTML으로 이동

순천향대

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

# 기상청 페이지 분석 - 테이블 시작 태그 (2)

- HTML 문서 위로 스크롤링하여 테이블의 시작 확인
  - <table class="table\_develop3"> 부분을 클릭하면 표 전체가 선택

순천향대

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

## 기상청 페이지 분석 - 테이블 HTML 분석 (1)

### □ 페이지의 서울을 클릭하여 관련 태그 및 테이블 분석

- <table>에서 도시 별 데이터는 <tr> 태그로 표시
- 데이터 값은 <td> 태그로 표시
  - <td> 태그 안에 <a> 태그에 도시 이름
  - 기온은 여섯 번째(인덱스 5) <td>의 값
  - 습도는 열 번째(인덱스 9) <td>의 값

The screenshot shows a weather table with columns: 지점 (Location), 날씨 (Weather), 시정 (Visibility), 풍향 (Wind Direction), 풍속 (Wind Speed), 풍향 (Wind Direction), 풍속 (Wind Speed), 풍향 (Wind Direction), 풍속 (Wind Speed), 풍향 (Wind Direction), 풍속 (Wind Speed), 풍향 (Wind Direction), 풍속 (Wind Speed). The table lists weather data for Seoul and other locations. The developer tools show the HTML structure, with red boxes highlighting the <tr> tag for Seoul and the <td> tags for temperature (25.6), wind speed (6.7), and humidity (30%).

순천향대학교 컴퓨터공학과

## 기상청 페이지 분석 - 테이블 HTML 분석 (2)

```
<table class="table_develop3" summary="기상실황표로 지점, 날씨, 기온, 강수, 바람, 기압등을 안내한 표입니다.">
...생략...
<tr>
  <td><a href="/weather/observation/currentweather.jsp?tm=2017.5.17.14:00&type=t99&mode=0&auto_man=m&stn=108">서울</a></td>
  <td>맑음</td>
  <td>18.9</td>
  <td>1</td>
  <td>1</td>
  <td>25.6</td>
  <td>6.7</td>
  <td>70</td>
  <td>&nbsp;</td>
  <td>30</td>
  <td>서남서</td>
  <td>2.1</td>
  <td>1010.1</td>
</tr>
...생략...
```

## 기상청 웹 크롤링 코드 (1)

```
import urllib.request
from bs4 import BeautifulSoup

# 기상청 현재 날씨 페이지 읽기
page =
    urllib.request.urlopen("http://www.weather.go.kr/weather/observation/currentweather.jsp
    ").read()
soup = BeautifulSoup(page, 'html.parser')    # 뷰티풀 수프 객체 생성

# 현재 날씨 테이블 검색
table = soup.find('table', { 'class': 'table_develop3' })
                                # 테이블 시작, <table class="table_develop3"> 검색
data = []                        # 데이터를 저장할 리스트 생성
```

## 기상청 웹 크롤링 코드 (2)

```
# 모든 <tr> 태그를 검색 각 도시의 데이터를 가져옴
for tr in table.find_all('tr'):    # 각 도시의 행 tr에 대해 반복
    tds = list(tr.find_all('td'))  # 모든 <td> 태그를 검색해 한 도시의 데이터(날씨) 리스트 생성
    for td in tds:                # 한 도시의 날씨 값들 td에 대해 반복
        if td.find('a'):          # <td> 안에 <a> 태그가 있으면(표 제목이 아니고 도시의 행인지 확인)
            point = td.find('a').text    # <a> 태그 안에서 도시를 가져옴
            temperature = tds[5].text    # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을 가져옴
            humidity = tds[9].text       # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를 가져옴
            data.append([point, temperature, humidity]) # data 리스트에 도시, 기온, 습도를 추가

print(data)                        # 리스트 data 출력

with open('weather.csv', 'w', encoding='utf8') as file: # weather.csv 파일을 쓰기 모드로 열기
    file.write('point,temperature,humidity\n')          # 컬럼 이름 추가
    for i in data:                                     # 리스트의 각 항목에 대해
        file.write('{0},{1},{2}\n'.format(i[0], i[1], i[2])) # 도시, 온도, 습도를 줄 단위로 저장
```



```
import urllib.request
from bs4 import BeautifulSoup

# 기상청 현재 날씨 페이지 읽기
page = urllib.request.urlopen("http://www.weather.go.kr/weather/observation/currentweather.jsp").read()
soup = BeautifulSoup(page, 'html.parser') # 뷰티풀 수프 객체 생성

# 현재 날씨 테이블 검색
table = soup.find('table', {'class': 'table_develop3'}) # 테이블 시작, <table class="table_develop3"> 검색
data = [] # 데이터를 저장할 리스트 생성

# 모든 <tr> 태그를 검색 각 도시의 데이터를 가져올
for tr in table.find_all('tr'): # 각 도시의 행 tr에 대해 반복
    tds = list(tr.find_all('td')) # 모든 <td> 태그를 검색해 한 도시의 데이터(날씨) 리스트 생성
    for td in tds: # 한 도시의 날씨 값을 td에 대해 반복
        if td.find('a'): # <a> 안에 <a> 태그가 있으면(표 제목이 아니고 도시의 행인지 확인)
            point = td.find('a').text # <a> 태그 안에서 지점을 가져올
            temperature = tds[5].text # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을 가져올
            humidity = tds[9].text # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를 가져올
            data.append([point, temperature, humidity]) # data 리스트에 지점, 기온, 습도를 추가

print(data) # 리스트 data 출력
```

```
[['서울', '27.1', '72'], ['백령도', '25.1', '73'], ['인천', '26.3', '72'], ['수원', '26.2', '80'], ['동두천', '26.4', '77'], ['파주', '27.7', '69'], ['강화', '27.5', '63'], ['양평', '25.3', '85'], ['이천', '27.7', '70'], ['북춘천', '27.9', '66'], ['북강릉', '27.7', '56'], ['울릉도', '28.0', '64'], ['속초', '30.9', '48'], ['철원', '26.6', '78'], ['대관령', '21.7', '79'], ['춘천', '28.8', '63'], ['강릉', '29.9', '49'], ['동해', '28.6', '52'], ['원주', '28.6', '53'], ['영월', '27.1', '58'], ['인제', '26.4', '62'], ['홍천', '25.0', '85'], ['태백', '26.9', '54'], ['정선군', '25.2', '64'], ['서산', '26.1', '92'], ['청주', '29.5', '59'], ['대전', '28.8', '65'], ['충주', '29.0', '61'], ['추풍령', '29.0', '48'], ['홍성', '28.1', '71'], ['제천', '26.6', '63'], ['보은', '28.9', '57'], ['천안', '27.2', '70'], ['보령', '28.8', '74'], ['부여', '28.6', '70'], ['금산', '29.6', '52'], ['전주', '28.7', '73'], ['광주', '28.7', '73'], ['목포', '29.6', '59'], ['여수', '28.2', '72'], ['흑산도', '27.3', '78'], ['군산', '28.2', '68'], ['완도', '29.7', '67'], ['고창', '29.0', '68'], ['순천', '29.2', '56'], ['부안', '29.9', '66'], ['임실', '28.3', '61'], ['정읍', '29.0', '63'], ['남원', '29.0', '57'], ['장수', '27.1', '57'], ['고창군', '29.5', '66'], ['영광군', '29.9', '68'], ['순창군', '29.1', '62'], ['보성군', '29.7', '64'], ['강진군', '29.6', '65'], ['장흥', '29.8', '63'], ['해남', '29.2', '70'], ['고흥', '29.1', '66'], ['광양시', '30.0', '55'], ['진도군', '29.3', '66'], ['제주', '30.7', '63'], ['고산', '28.5', '77'], ['성산', '29.7', '67'], ['서귀포', '28.6', '91'], ['안동', '29.4', '56'], ['포항', '31.7', '53'], ['대구', '32.8', '35'], ['울산', '30.4', '53'], ['창원', '31.1', '50'], ['부산', '29.0', '68'], ['울진', '27.5', '70'], ['삼주', '30.1', '58'], ['통영', '29.3', '65'], ['진주', '30.8', '56'], ['김해시', '31.3', '65'], ['북창원', '30.1', '55'], ['양산시', '32.0', '54'], ['의령군', '32.3', '50'], ['함양군', '30.2', '60'], ['봉화', '28.0', '58'], ['영주', '28.6', '58'], ['문경', '29.4', '55'], ['청송군', '30.3', '57'], ['영덕', '30.4', '58'], ['의성', '32.1', '38'], ['구미', '30.6', '54'], ['영천', '31.7', '45'], ['경주시', '32.1', '57'], ['거창', '30.3', '47'], ['합천', '31.5', '45'], ['밀양', '32.0', '41'], ['산청', '30.1', '60'], ['거제', '30.0', '66'], ['남해', '31.3', '53']]
```

```
with open('weather.csv', 'w', encoding='utf8') as file: # weather.csv 파일을 쓰기 모드로 열기
    file.write('point,temperature,humidity\n') # 컬럼 이름 추가
    for i in data: # data를 반복하면서
        file.write('{0},{1},{2}\n'.format(i[0], i[1], i[2])) # 지점, 온도, 습도를 줄 단위로 저장
```

point,temperature,humidity

```
서울,27.1,72
백령도,25.1,73
인천,26.3,72
수원,26.2,80
동두천,26.4,77
파주,27.7,69
강화,27.5,63
양평,25.3,85
이천,27.7,70
북춘천,27.9,66
북강릉,27.7,56
울릉도,28.0,64
속초,30.9,48
철원,26.6,78
대관령,21.7,79
춘천,28.8,63
강릉,29.9,49
동해,28.6,52
원주,28.6,53
영월,27.1,58
인제,26.4,62
홍천,25.0,85
태백,26.9,54
정선군,25.2,64
서산,26.1,92
청주,29.5,59
대전,28.8,65
충주,29.0,61
추풍령,29.0,48
홍성,28.1,71
제천,26.6,63
보은,28.9,57
천안,27.2,70
보령,28.8,74
```

Select items to perform actions on them.

- 0 /
- bs-test.ipynb
- jupyter-test.ipynb
- weather.ipynb
- jupyter-test.py
- weather.csv

1. “기상청 웹 크롤링”을 작성하고 실행
2. “기상청 웹 크롤링”을 아래와 같이 수정하여 작성하고 실행

- 파일의 첫 라인에 아래와 같이 현재 시간을 출력
 

```
2019년 08월 19일 11시 00분 현재 날씨
도시, 온도, 습도
서울, 26.4, 62
백령도, 24.3, 72
인천, 25.6, 74
스위 26.9, 60
```

  - 2019년 08월 19일 11시 00분 현재 날씨
- 힌트
  - 현재 시간을 입력하는 다음 태그 라인 검색
    - `<input type="text" id="observation_text" class="time" name="tm" value="2019.08.19.09:00">`
    - 검색된 data에 value 속성값 만 추출, `data['value']`
    - 추출된 시간 문자열에 `split()` 메서드 사용하여 년,월,일,시간,분 분리

### 3. 앞에서 배운 내용을 사용한 임의의 프로그램 작성

- 임의의 웹 페이지 접속하여 웹 크롤링
- 프로그램 소스, 프로그램 설명
- 실행 결과