

프로젝트 진행 순서

1. 프로젝트 개요
2. 데이터 수집
3. 데이터 정제 및 가공
4. 데이터 시각화

[프로젝트 목적]

전 세계 개발자를 대상으로 설문 조사를 한 결과를 분석하는 프로젝트를 진행하면서 취합된 데이터를 집계 형식으로 변경하는 방식을 이해하고 목적에 맞는 그래프를 선택한다. 그래프를 수직/수평 막대그래프로 변경하고, 파이 그래프를 그릴 수 있다. 그래프 크기 변경, 정렬 방식 변경, 백분율 표기 등 기본 그래프를 편집할 수 있다.

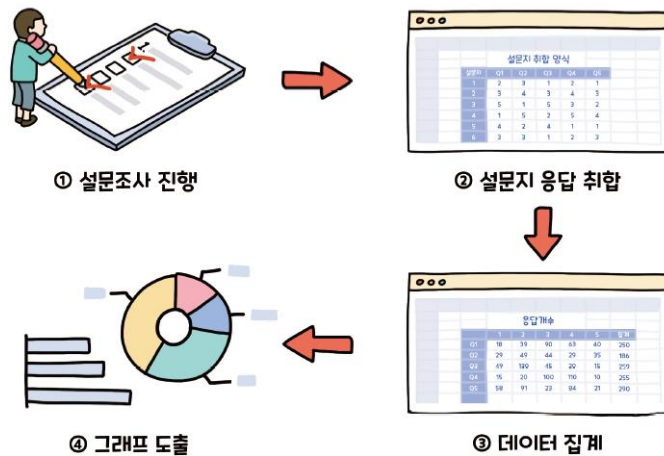
Section 01

프로젝트 개요

1. 프로젝트의 목표

■ 데이터 취합과 집계

✓ 데이터를 취합하고 집계하는 과정



설문 조사 결과 수집 과정

2. 문제 정의

■ 글로벌 소프트웨어 엔지니어가 되기 위한 전략 세우기

- 개발자가 많은 나라와 현재 유행하고 있는 기술 파악하기
 - 데이터 : 전 세계 소프트웨어 엔지니어들이 참여한 커뮤니티 활용하기

안녕하세요, 저는 글로벌 소프트웨어 엔지니어가 되는 것이 꿈인 신후입니다. 해외에 진출하기 위해 코딩 실력을 쌓고 영어 공부를 하는 것도 중요하지만, 어떤 나라를 선택하고 어느 시점에 이동하는 것이 가장 좋을지 고민입니다.

소프트웨어 엔지니어가 많은 나라일수록 취업 기회가 더 많을 것 같습니다. 그리고 제가 어떤 기술을 배워야 할지 알기 위해 연령 대별로 현재 소프트웨어 엔지니어가 사용하는 프로그래밍 언어의 종류도 알고 싶습니다. 이에 대한 데이터를 수집해 제가 배운 파이썬 데이터 분석 기술로 여러 그래프를 그려 확인하고 싶은데, 어떻게 해야 하나요?



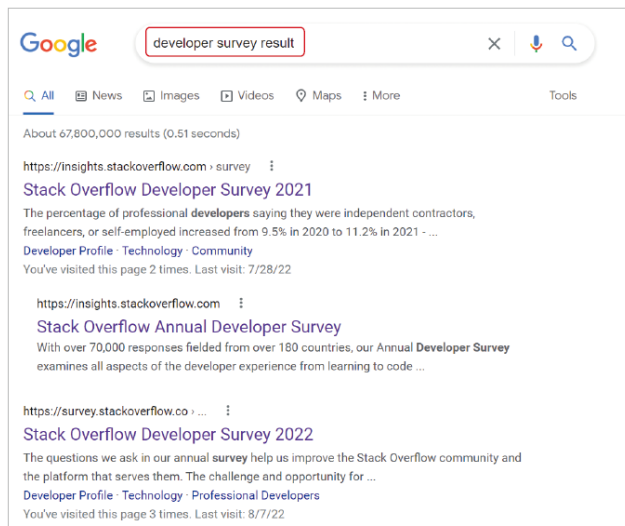
Section 02

데이터 수집

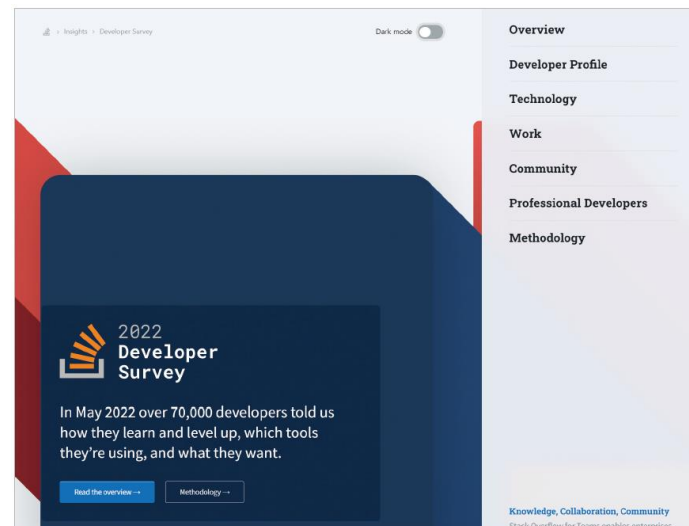
1. 데이터 조사 및 확인

■ 개발자 설문 조사 결과 검색

- 구글에서 'developer survey result'를 검색
 - [그림 5-2]와 같이 검색 결과 상단에서 스택오버플로 웹 사이트를 확인
- 설문 조사 개요 페이지 방문
 - [그림 5-2]에서 세 번째로 보이는 2022년 결과를 클릭 → 조사 결과를 취합한 그래프 확인



개발자 설문 조사 구글 검색 결과



2022년 스택오버플로 개발자 설문 조사 결과
(<https://survey.stackoverflow.co/2022>)

1. 데이터 조사 및 확인

■ 참고 대상 그래프 확인

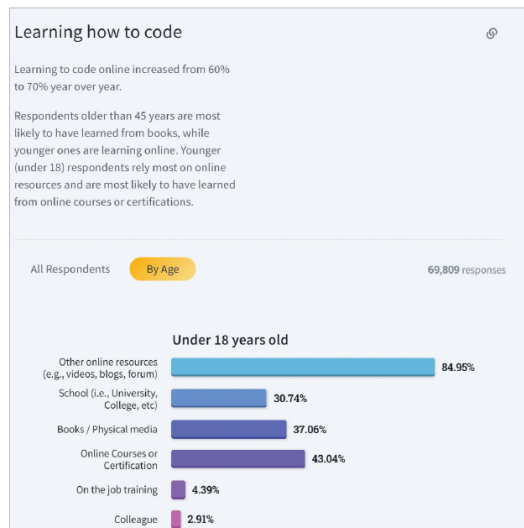
- 연령대별 프로그래밍 학습 방법 그래프

→ 신후가 앞으로 어떻게 프로그래밍을 학습하면 좋을지 알아보는 데 참고할 수 있는 정보

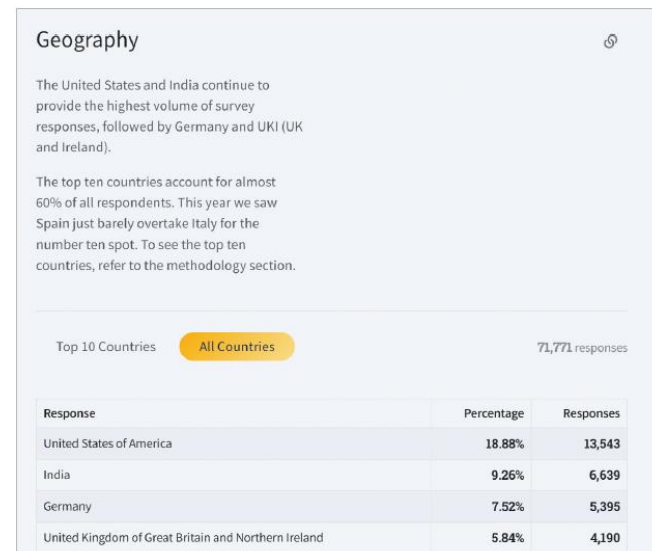
- 개발자들의 거주 국가 비율

→ 이 정보는 국가별 개발자의 수 가 아님

→ 스택오버플로를 사용하면서 설문 조사에 응한 개발자 수를 토대로 함



연령대별 프로그래밍 학습 방법



개발자 거주 국가 비율

1. 데이터 조사 및 확인

■ 참고 대상 그래프 확인

- 프로그래밍 언어 선호 비율 그래프

→ 정의된 문제에서 구하고자 하는 데이터 조건에 부합함

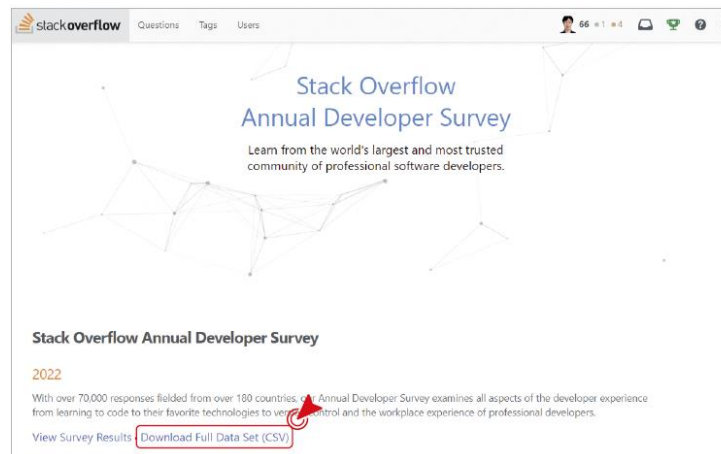


프로그래밍 언어 선호 비율

2. 데이터 다운로드 및 탑재 준비

■ 개발자 설문 조사 데이터 다운로드

- 스택오버플로 설문 조사 인사이트 페이지 (<https://insights.stackoverflow.com/survey>)
 - <Download Full Data Set(CSV)> 링크를 클릭 → zip 파일 다운로드 → 압축해제
- ① README_2022.txt: 다운로드한 파일들에 대한 설명
- ② so_survey_2022.pdf: 설문 조사 질문지
- ③ survey_results_public.csv: 취합 결과
- ④ survey_results_schema.csv: so_survey_2022.pdf 파일을 웹으로 표현한 파일

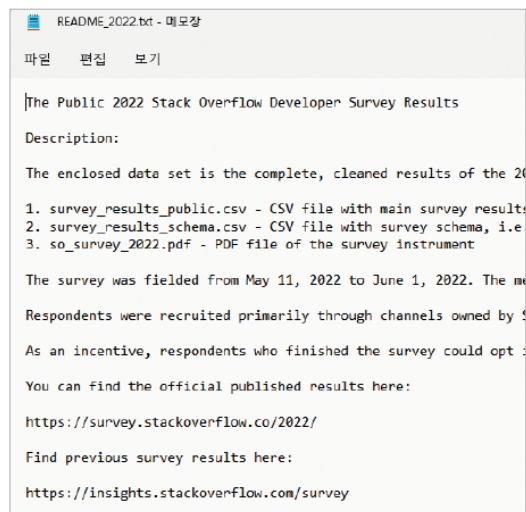


2022년 개발자 설문 조사 데이터 다운로드

2. 데이터 다운로드 및 탑재 준비

■ 설문 조사 데이터의 종류

- README_2022.txt
 - 다운로드한 파일과 설문 조사에 대한 자세한 설명을 영문으로 확인할 수 있음
- so_survey_2022.pdf
 - 설문 조사의 질문지를 확인할 수 있음
 - ✓ 데이터를 살펴보기 전에 어떤 질문에 대한 답인지 이해하기 위해서 반드시 확인해야 함



README_2022.txt 파일의 내용

A screenshot of a survey question from a PDF file. The question is 'At what age did you write your first line of code or program? (e.g., webpage, Hello World, Scratch project)'. Below the question are eight radio button options representing age ranges: 'Younger than 5 years', '5 - 10 years', '11 - 17 years', '18 - 24 years', '25 - 34 years', '35 - 44 years', '45 - 54 years', '55 - 64 years', and 'Older than 64 years'.

설문 조사 질문의 예시

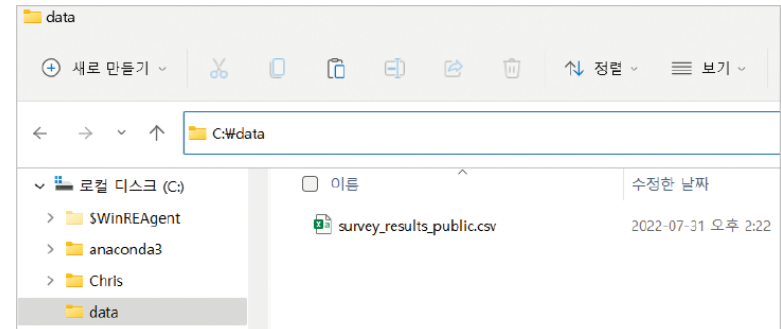
2. 데이터 다운로드 및 탑재 준비

■ 설문 조사 데이터의 종류

- survey_results_public.csv
 - 실제로 취합한 데이터
 - CSV 포맷으로 쉼표 기호(,)로 항목이 구분된 응답 내용이 한 줄에 한 사람씩 나열되어 있음
 - ✓ 이 파일을 주피터 노트북에 탑재하기 위해 C:/data 폴더에 복사→ 붙여 넣기



설문 조사 취합 결과 확인하기



데이터 옮기기(C:\data\survey_results_public.csv)

3. 데이터 읽어오기

■ 신규 주피터 노트북 파일 생성 및 CSV 파일 업로드

- 신규 주피터 노트북 파일을 생성 :
- 다음 소스 코드를 실행해 CSV 파일을 주피터 노트북에 업로드

In [1]:	<pre># 판다스 라이브러리 탑재 import pandas as pd # CSV 파일 읽어오기 raw_data = pd.read_csv(".\survey_results_public.csv") # 데이터 확인하기 raw_data.info()</pre>
Out [1]:	<pre><class 'pandas.core.frame.DataFrame'> RangeIndex: 73268 entries, 0 to 73267 Data columns (total 79 columns): # Column Non-Null Count Dtype --- - 0 ResponseId 73268 non-null int64 1 MainBranch 73268 non-null object 2 Employment 71709 non-null object 3 RemoteWork 58958 non-null object</pre>

3. 데이터 읽어오기

■ 신규 주피터 노트북 파일 생성 및 CSV 파일 업로드

Out [1]:	4 CodingActivities	58899 non-null object
	----- 중략 -----	
	73 TrueFalse_1	35819 non-null object
	74 TrueFalse_2	35715 non-null object
	75 TrueFalse_3	35749 non-null object
	76 SurveyLength	70444 non-null object
	77 SurveyEase	70508 non-null object
	78 ConvertedCompYearly	38071 non-null float64
	dtypes: float64(5), int64(1), object(73)	
	memory usage: 44.2+ MB	

Section 03

데이터 정제 및 가공

1. 데이터 필터링

■ 일부 데이터 확인하기

- Out[2]: 결과 설문 조사의 각 질문은 열로, 설문에 대한 응답은 행으로 구분됨

In [2]:

일부 데이터 확인하기
raw_data.head()

Out [2]:

	ResponseId	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	LearnCode
0	1	None of these	NaN	NaN	NaN	NaN	NaN	
1	2	I am a developer by profession	Employed, full-time	Fully remote	Hobby; Contribute to open-source projects	NaN	NaN	
2	3	I am not primarily a developer, but I write code...	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media; Friend or family member...	Documentation, Blogs, Forums
3	4	I am a developer by profession	Employed, full-time	Fully remote	I don't code outside of work	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Books / Physical media; School (i.e., University...)	
4	5	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Other online resources (e.g., videos, blogs, forums, etc.)	Documentation, Blogs, Forums

5 rows x 79 columns

1. 데이터 필터링

■ 필요한 열만 추출하기

- Age: 나이
- Country: 국가 정보
- LanguageHaveWorkedWith: 다룰 줄 아는 프로그래밍 언어
- LearnCode: 프로그래밍을 배운 방법

In [3]:

```
# 필요한 열 추출
revised_data = raw_data[["Age", "Country"
                          "LanguageHaveWorkedWith", "LearnCode"]]

# 일부 값 확인
revised_data.head()
```

Out [3]:

	Age	Country	LanguageHaveWorkedWith	LearnCode
0	NaN	NaN	NaN	NaN
1	NaN	Canada	JavaScript;TypeScript	NaN
2	25-34 years old	United Kingdom of Great Britain and Northern I...	C#;C++;HTML/CSS;JavaScript;Python	Books / Physical media;Friend or family member...
3	35-44 years old	Israel	C#;JavaScript;SQL;TypeScript	Books / Physical media;School (i.e., Universit...
4	25-34 years old	United States of America	C#;HTML/CSS;JavaScript;SQL;Swift;TypeScript	Other online resources (e.g., videos, blogs, f...

2. 특정 데이터 탐색하기

■ 특정 열 데이터 조회하기

- 설문 조사에 응한 개발자의 연령대
 - 다음 소스 코드를 입력 → Age 열에 있는 데이터를 확인
 - 73,268개의 데이터에 응답자가 선택한 연령대 값 도출

In [4]:	# 연령대 정보 확인하기 revised_data["Age"]
Out [4]:	0 NaN 1 NaN 2 25-34 years old 3 35-44 years old 4 25-34 years old ... 73263 25-34 years old 73264 25-34 years old 73265 55-64 years old 73266 55-64 years old 73267 25-34 years old Name: Age, Length: 73268, dtype: object

2. 특정 데이터 탐색하기

■ 유일 데이터 확인하기

- 응답자들이 선택한 연령대의 종류를 확인하기
 - `drop_duplicates()` 메서드를 실행 → 중복된 값을 제거
 - 총 9종류의 응답 유형이 있는 것을 확인

In [5]:	# 중복 값 삭제하기 <code>revised_data["Age"].drop_duplicates()</code>
Out [5]:	0 NaN 2 25-34 years old 3 35-44 years old 6 Under 18 years old 7 18-24 years old 9 45-54 years old 24 55-64 years old 77 65 years or older 451 Prefer not to say Name: Age, dtype: object

3. 데이터 그룹화하기

■ 데이터 그룹화 객체 만들기

- groupby() 메서드
 - 판다스 데이터프레임의 값을 집계하기 위해 데이터를 그룹화하는 메서드
 - 다음 소스 코드를 실행하면 DataFrameGroupBy 객체가 반환됨

In [6]:	# 연령대 데이터 그룹화하기 revised_data.groupby(["Age"])
Out [6]:	<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000232ECBDD700>

3. 데이터 그룹화하기

■ 호출 가능한 메서드 확인하기

- DataFrameGroupBy 객체에서 사용할 수 있는 메서드의 종류를 확인하기
 - 다음과 같이 소스 코드를 입력하고 마침표 기호(.)를 입력
 - <Tab> 키를 누르고 잠시 기다리면 호출할 수 있는 메서드의 리스트가 나타남



```
In [ ]: revised_data.groupby(["Age"]).|
```

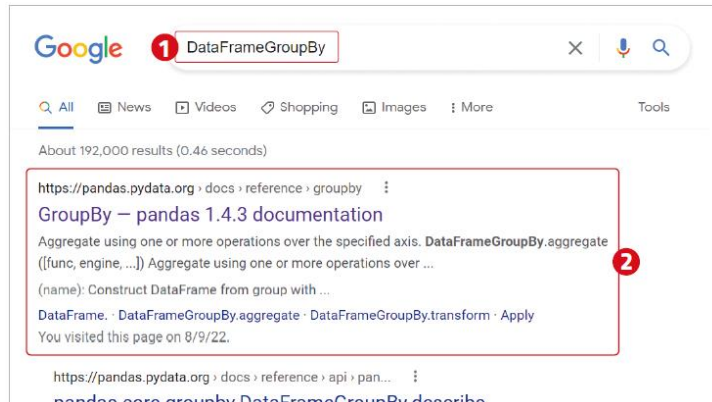
The dropdown menu lists the following methods: .agg, .aggregate, .all, .any, .apply, .as_index, .axis, .backfill, .bfill, .boxplot.

호출 가능한 메서드 확인하기

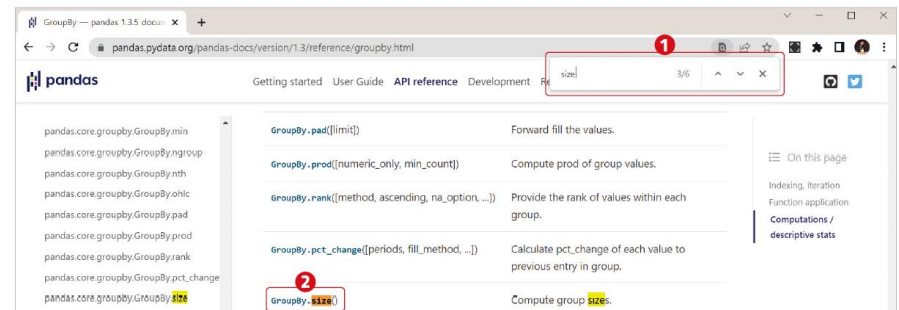
3. 데이터 그룹화하기

■ 공식 레퍼런스 문서 확인하기

- GroupBy.size() 메서드
 - 그룹화된 객체에서 각 그룹의 크기를 반환하는 메서드
- 구글에서 'DataFrameGroupBy'를 검색 → 판다스의 레퍼런스 문서 클릭
 - 오른쪽 상단에서 버전 정보를 현재 사용하고 있는 버전(1.4.2)을 확인
 - 메서드 리스트를 확인 → <Ctrl> + <F>으로 size를 입력 → GroupBy.size() 메서드 클릭



DataFrameGroupBy 레퍼런스 문서 검색하기

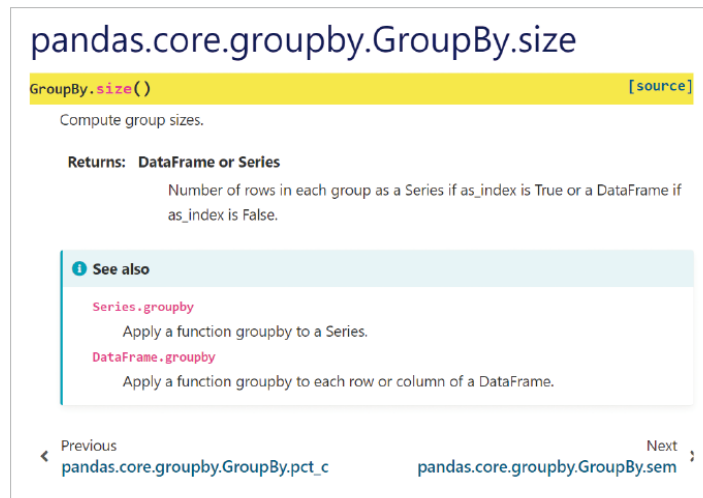


판다스 문서에서 size 검색하기

3. 데이터 그룹화하기

■ 공식 레퍼런스 문서 확인하기

- `GroupBy.size()` 메서드 상세 설명
 - 그룹의 크기를 계산하는 기능을 한다는 것을 알 수 있음



`GroupBy.size()` 메서드 상세 설명

4. 데이터 그룹별 크기 구하기

■ 연령대별 응답 숫자 확인하기

- GroupBy.size() 메서드 사용

→ 그룹화 과정 : 데이터 값이었던 연령대 정보가 색인으로 이동해 그룹을 형성하고 있음

In [7]:	# 연령대 그룹별 크기 확인하기 size_by_age = revised_data.groupby(["Age"]).size() size_by_age	
Out [7]:	Age 18-24 years old 16646 25-34 years old 28112 35-44 years old 13988 45-54 years old 5281 55-64 years old 1978 65 years or older 554 Prefer not to say 521 Under 18 years old 3866 dtype: int64	

4. 데이터 그룹별 크기 구하기

■ 국가별 응답 숫자 확인하기

- 국가별 응답 숫자 확인 : 열 이름 변경
 - 180개국의 정보가 색인으로 이동하여 응답자 수가 값으로 출력

In [8]:	# 국가 그룹별 크기 확인하기 size_by_country = revised_data.groupby(["Country"]).size() size_by_country	
Out [8]:	Country Afghanistan 69 Albania 59 Algeria 36 Andorra 15 Angola 12 ... Venezuela, Bolivarian Republic of... 119 Viet Nam 317 Yemen 14 Zambia 20 Zimbabwe 30 Length: 180, dtype: int64	

Section 04

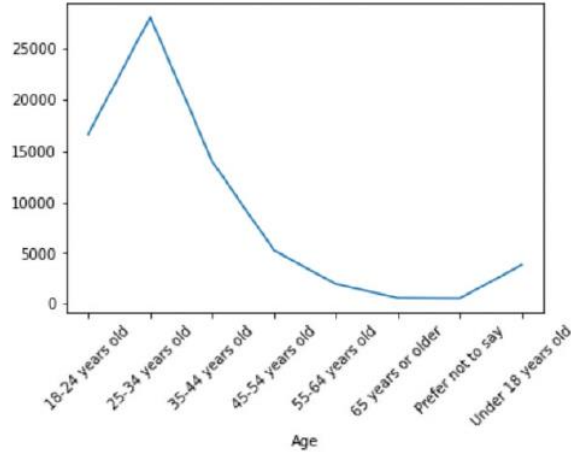
데이터 시각화

1. 연령대별 응답자 수 그래프 그리기

■ 선 그래프

- plot() 혹은 plot.line() 메서드

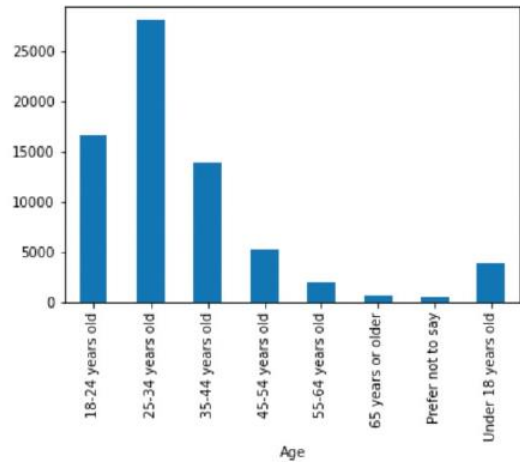
→ 연령대를 의미하는 행 이름 → X축 각도 45도 비틀기

In [9]:	<pre># 선 그래프 그리기 size_by_age.plot.line(rot=45)</pre>																		
Out [9]:	<p><AxesSubplot: xlabel='Age'></p>  <table border="1"><thead><tr><th>Age</th><th>Count (approx.)</th></tr></thead><tbody><tr><td>18-24 years old</td><td>17,000</td></tr><tr><td>25-34 years old</td><td>27,000</td></tr><tr><td>35-44 years old</td><td>14,000</td></tr><tr><td>45-54 years old</td><td>5,000</td></tr><tr><td>55-64 years old</td><td>2,000</td></tr><tr><td>65 years or older</td><td>1,000</td></tr><tr><td>Prefer not to say</td><td>1,000</td></tr><tr><td>Under 18 years old</td><td>4,000</td></tr></tbody></table>	Age	Count (approx.)	18-24 years old	17,000	25-34 years old	27,000	35-44 years old	14,000	45-54 years old	5,000	55-64 years old	2,000	65 years or older	1,000	Prefer not to say	1,000	Under 18 years old	4,000
Age	Count (approx.)																		
18-24 years old	17,000																		
25-34 years old	27,000																		
35-44 years old	14,000																		
45-54 years old	5,000																		
55-64 years old	2,000																		
65 years or older	1,000																		
Prefer not to say	1,000																		
Under 18 years old	4,000																		

1. 연령대별 응답자 수 그래프 그리기

■ 수직 막대그래프

- plot.bar() 메서드
 - 연령대가 낮은 그룹에서 높은 그룹 순으로 나열
 - 연령대 정보를 y축으로 옮기고 응답 숫자를 x축에 표현하는 수평 막대그래프 생성

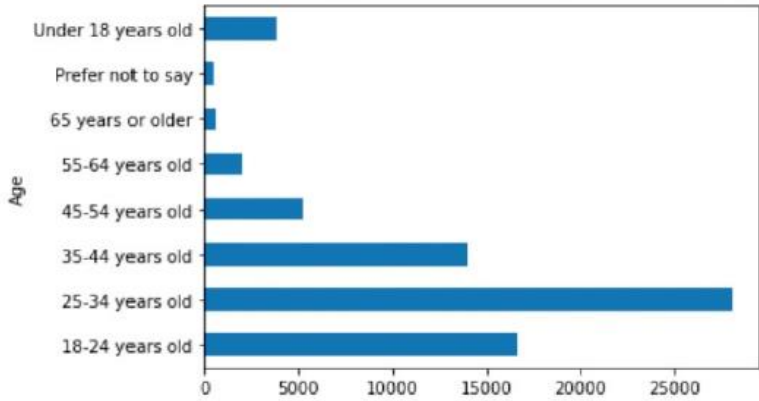
In [10]:	# 수직 막대그래프 그리기 size_by_age.plot.bar()																		
Out [10]:	<p><AxesSubplot: xlabel='Age'></p>  <table border="1"><thead><tr><th>Age</th><th>Count</th></tr></thead><tbody><tr><td>18-24 years old</td><td>16500</td></tr><tr><td>25-34 years old</td><td>28000</td></tr><tr><td>35-44 years old</td><td>14000</td></tr><tr><td>45-54 years old</td><td>5000</td></tr><tr><td>55-64 years old</td><td>2000</td></tr><tr><td>65 years or older</td><td>500</td></tr><tr><td>Prefer not to say</td><td>500</td></tr><tr><td>Under 18 years old</td><td>4000</td></tr></tbody></table>	Age	Count	18-24 years old	16500	25-34 years old	28000	35-44 years old	14000	45-54 years old	5000	55-64 years old	2000	65 years or older	500	Prefer not to say	500	Under 18 years old	4000
Age	Count																		
18-24 years old	16500																		
25-34 years old	28000																		
35-44 years old	14000																		
45-54 years old	5000																		
55-64 years old	2000																		
65 years or older	500																		
Prefer not to say	500																		
Under 18 years old	4000																		

1. 연령대별 응답자 수 그래프 그리기

■ 수평 막대그래프

- plot.barh() 메서드

→ x축과 y축을 바꾼 수평 막대그래프 생성

In [11]:	<pre># 수평 막대그래프 그리기 size_by_age.plot.barh()</pre>																		
Out [11]:	<p><AxesSubplot:ylabel='Age'></p>  <table border="1"><thead><tr><th>Age</th><th>Count (approx.)</th></tr></thead><tbody><tr><td>Under 18 years old</td><td>4,000</td></tr><tr><td>Prefer not to say</td><td>500</td></tr><tr><td>65 years or older</td><td>500</td></tr><tr><td>55-64 years old</td><td>2,000</td></tr><tr><td>45-54 years old</td><td>5,000</td></tr><tr><td>35-44 years old</td><td>14,000</td></tr><tr><td>25-34 years old</td><td>27,000</td></tr><tr><td>18-24 years old</td><td>16,000</td></tr></tbody></table>	Age	Count (approx.)	Under 18 years old	4,000	Prefer not to say	500	65 years or older	500	55-64 years old	2,000	45-54 years old	5,000	35-44 years old	14,000	25-34 years old	27,000	18-24 years old	16,000
Age	Count (approx.)																		
Under 18 years old	4,000																		
Prefer not to say	500																		
65 years or older	500																		
55-64 years old	2,000																		
45-54 years old	5,000																		
35-44 years old	14,000																		
25-34 years old	27,000																		
18-24 years old	16,000																		

1. 연령대별 응답자 수 그래프 그리기

■ 그룹 이름(색인) 순서 변경하기

- Out[11]:의 실행 결과 : 그룹 이름이 문자 우선순위로 정렬됨

→연령대가 가장 낮은 Under 18 years old 그룹이 제일 위에, 값이 없거나 응답을 거부한 Prefer not to say 그룹이 제일 아래에 위치하도록 조정

- 현재 색인 정보 확인하기

In [12]:	# 현재 색인 확인하기 size_by_age.index
Out [12]:	Index(['18-24 years old', '25-34 years old', '35-44 years old', '45-54 years old', '55-64 years old', '65 years or older', 'Prefer not to say', 'Under 18 years old'], dtype='object', name='Age')

1. 연령대별 응답자 수 그래프 그리기

■ 그룹 이름(색인) 순서 변경하기

- `reindex()` 메서드

→ 색인 조정 메서드를 사용하여 색인 순서를 그래프 하단에서 상단 순으로 나열

```
In [13]: # 재색인하기
reindexed_size_by_age = size_by_age.reindex(index=[
    'Prefer not to say',
    '65 years or older',
    '55-64 years old',
    '45-54 years old',
    '35-44 years old',
    '25-34 years old',
    '18-24 years old',
    'Under 18 years old'
])

# 재색인한 결과 확인하기
reindexed_size_by_age
```

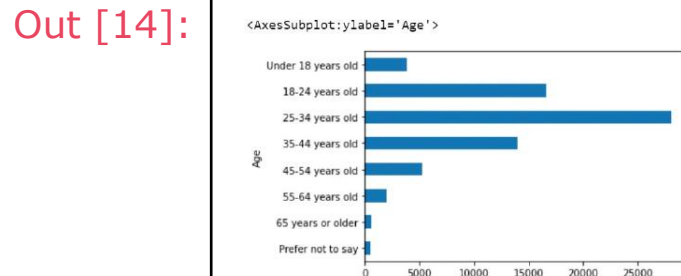
1. 연령대별 응답자 수 그래프 그리기

■ 그룹 이름(색인) 순서 변경하기

```
Out [13]: Age
Prefer not to say      521
65 years or older      554
55-64 years old        1978
45-54 years old        5281
35-44 years old        13988
25-34 years old        28112
18-24 years old        16646
Under 18 years old      3866
dtype: int64
```

- 재색인한 데이터프레임으로 수평 막대그래프 생성

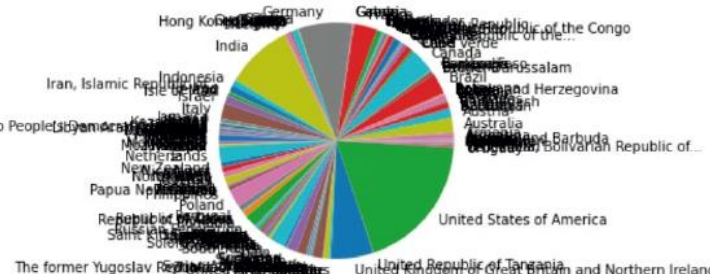
```
In [14]: # 수평 그래프 다시 그리기
reindexed_size_by_age.plot.barh()
```



2. 국가별 응답자 수 그래프 그리기

■ 기본 파이 그래프

- plot.pie() 메서드

In [15]:	# 기본 파이 그래프 그리기 size_by_country.plot.pie()
Out [15]:	<AxesSubplot:ylabel='None'> 

→ 기본 설정 그대로 파이 그래프를 생성하니 다음과 같은 단점이 보임

- ✓ 원이 너무 작다.
- ✓ 나라가 너무 많아서 이름이 겹쳐 출력되어 식별이 불가능하다.
- ✓ 파이가 너무 작게 쪼개져서 크기를 비교하기 어렵다.

2. 국가별 응답자 수 그래프 그리기

■ 그래프 크기 조정하기

- 그래프 크기를 조정하기 위해 판다스 공식 문서를 확인
- `pie()` 메서드 인수로 `**kwargs` 확인
 - 키와 값으로 이루어진 딕셔너리를 의미
 - ✓ 자세한 키 리스트와 사용법은 `<DataFrame.plot()>` 링크에서 확인 가능
- `DataFrame.plot()` 레퍼런스 문서에 있는 인수를 그대로 사용함
 - 데이터프레임 `plot` 객체에서 호출하는 모든 메서드에서 공통으로 사용할 수 있다는 의미임

pandas.DataFrame.plot.pie [\[source\]](#)

`DataFrame.plot.pie(**kwargs)`

Generate a pie plot.

A pie plot is a proportional representation of the numerical data in a column. This function wraps `matplotlib.pyplot.pie()` for the specified column. If no column reference is passed and `subplots=True` a pie plot is drawn for each numerical column independently.

Parameters: `y` : *int or label, optional*
Label or position of the column to plot. If not provided, `subplots=True` argument must be passed.

****kwargs**
Keyword arguments to pass on to `DataFrame.plot()`.

판다스 파이 메서드 레퍼런스 문서 확인하기

pandas.DataFrame.plot [\[source\]](#)

`DataFrame.plot(*args, **kwargs)`

Make plots of Series or DataFrame.

Uses the backend specified by the option `plotting.backend`. By default, matplotlib is used.

(rows, columns, ...)

figsize : *a tuple (width, height) in inches*
Size of a figure object.

plot 메서드 인수 리스트

2. 국가별 응답자 수 그래프 그리기

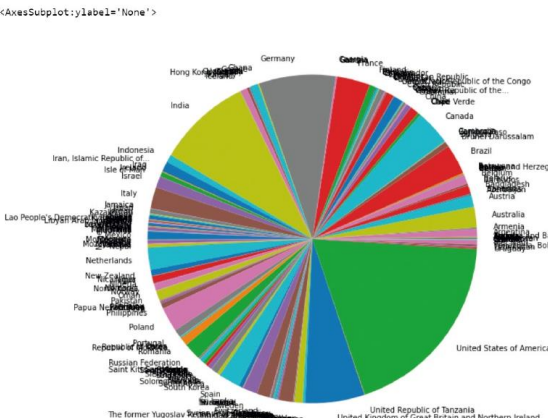
■ 그래프 크기 조정하기

- 인수 figsize 내용 확인하기

→ 인수 figsize에 인치로 표기한 너비(width), 높이(height) 로 구성된 튜플을 넣는다는 설명

figsize : a tuple (width, height) in inches

- 파이 그래프의 너비와 높이 10인치로 조정하기

In [16]:	# 파이 그래프 크기 조정하기 size_by_country.plot.pie(figsize=(10, 10))
Out [16]:	<p><AxesSubplot:ylabel='None'></p> 

2. 국가별 응답자 수 그래프 그리기

■ 상위 20개만 표현하기

- nlargest() 메서드

→ 데이터프레임에서 상위 N개 데이터를 추출하는 메서드

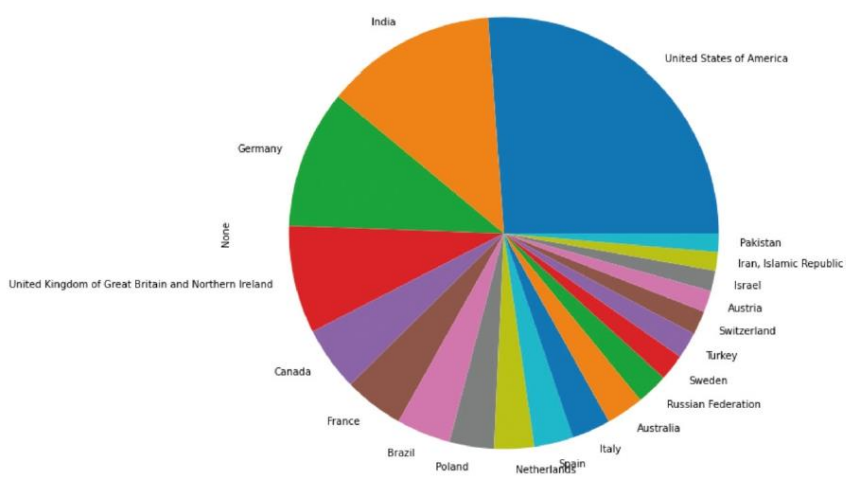
In [17]:	# 국가 상위 20개 확인하기 size_by_country.nlargest(20)
Out [17]:	Country7 United States of America 13543 India 6639 Germany 5395 United Kingdom of Great Britain and Northern Ireland 4190 Canada 2490 France 2328 -----중략----- Austria 833 Israel 795 Iran, Islamic Republic of... 723 Pakistan 711 dtype: int64

2. 국가별 응답자 수 그래프 그리기

■ 상위 20개 데이터 파이 그래프 그리기

- 국가 정보 식별

→ 미국 > 인도 > 독일 > 영국 > 캐나다 순의 국가별 응답자 수 확인

In [18]:	<pre># 국가 상위 20개 파이 그래프 그리기 size_by_country.nlargest(20).plot.pie(figsize=(10, 10))</pre>
Out [18]:	<p><AxesSubplot:ylabel='None'></p> 

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 특정 열 추출하기

- LanguageHaveWorkedWith

→ 각 응답자가 직접 사용한 경험이 있는 프로그래밍 언어를 나열한 열

→ Out [19]:의 출력 결과 : 세미클론(;)으로 구분된 프로그래밍 언어가 나열

→ dtype: object를 통해 이 값이 하나의 객체인 것을 알 수 있음

In [19]:	<pre># 프로그래밍 언어 데이터 추출 languages = revised_data["LanguageHaveWorkedWith"] # 데이터 확인 languages</pre>
Out [19]:	<pre>0 NaN 1 JavaScript;TypeScript # ;으로 값을 구분 2 C#;C++;HTML/CSS;JavaScript;Python 하여 나열함 3 C#;JavaScript;SQL;TypeScript 4 C#;HTML/CSS;JavaScript;SQL;Swift;TypeScript ... 73265 HTML/CSS;JavaScript;PHP;Python;SQL 73266 C#;Delphi;VBA 73267 C#;JavaScript;Lua;PowerShell;SQL;TypeScript Name: LanguageHaveWorkedWith, Length: 73268, dtype: object</pre>

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 단일 객체를 리스트로 변환하기

- split() 메서드

→ 프로그래밍 언어별로 그룹화를 하려면 각 프로그래밍 언어를 하나의 값으로 쪼개야 함

→ 이럴때 문자열에서 사용할 수 있는 split() 메서드

→ split() 메서드의 인수에는 값을 구분할 수 있는 구분자가 들어감

In [20]:	<pre># 데이터 문자열 변환 후 구분자(;)로 구분 languages = languages.str.split(";") # 데이터 확인 languages</pre>
Out [20]:	<pre>0 NaN 1 [JavaScript, TypeScript] 2 [C#, C++, HTML/CSS, JavaScript, Python] 3 [C#, JavaScript, SQL, TypeScript] 4 [C#, HTML/CSS, JavaScript, SQL, Swift, TypeScr... ... 73265 [HTML/CSS, JavaScript, PHP, Python, SQL] 73266 [C#, Delphi, VBA] 73267 [C#, JavaScript, Lua, PowerShell, SQL, TypeScr... Name: LanguageHaveWorkedWith, Length: 73268, dtype: object</pre>

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 리스트의 각 항목을 열로 나누기

- explode() 메서드

→ 리스트의 항목을 행으로 나누는 메서드

In [21]:	# 리스트 항목을 행으로 나누기 exploded_languages = languages.explode() # 데이터 확인 exploded_languages	
Out [21]:	0	NaN
	1	JavaScript
	1	TypeScript
	2	C#
	2	C++
		...
	73267	JavaScript
	73267	Lua
	73267	PowerShell
	73267	SQL
	73267	TypeScript
	Name: LanguageHaveWorkedWith, Length: 370114, dtype: object	

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 프로그래밍별 응답 수 확인하기

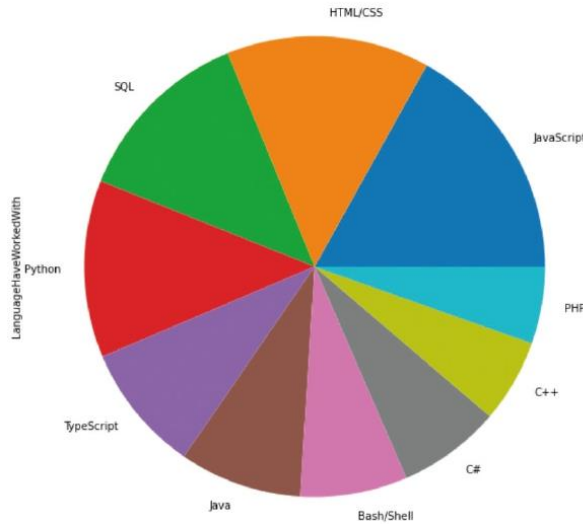
- explode() 메서드
 - 앞 절에서 국가별 응답자 수 그래프를 그릴 때 다룬 데이터와 동일한 형태가 되었음
 - 그래프에서와 동일한 방식으로 그룹화하여 각 그룹의 크기로 이루어진 데이터로 변환

In [22]:	<pre># 프로그래밍별 응답 수 구하기 size_by_languages = exploded_languages.groupby(exploded_languages).size() # 데이터 확인 size_by_languages</pre>
Out [22]:	<pre>LanguageHaveWorkedWith APL 504 Assembly 3887 Bash/Shell 20656 C 13692 -----< 중략 >----- Scala 1837 Solidity 1031 Swift 3489 TypeScript 24752 VBA 3185 Name: LanguageHaveWorkedWith, dtype: int64</pre>

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 프로그래밍별 응답 수 확인하기

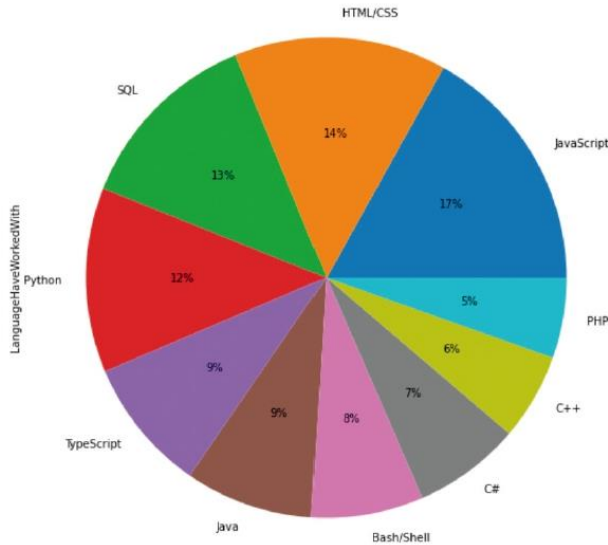
- 상위 10개 프로그래밍 언어를 파이 그래프로 그리기

In [23]:	<pre># 파이 그래프 그리기 size_by_languages.nlargest(10).plot.pie(figsize=(10, 10))</pre>																						
Out [23]:	<pre><AxesSubplot:ylabel='LanguageHaveWorkedWith'></pre>  <table border="1"><thead><tr><th>Language</th><th>Proportion (approx.)</th></tr></thead><tbody><tr><td>JavaScript</td><td>18%</td></tr><tr><td>HTML/CSS</td><td>15%</td></tr><tr><td>SQL</td><td>12%</td></tr><tr><td>Python</td><td>10%</td></tr><tr><td>TypeScript</td><td>8%</td></tr><tr><td>Java</td><td>7%</td></tr><tr><td>Bash/Shell</td><td>6%</td></tr><tr><td>C#</td><td>5%</td></tr><tr><td>C++</td><td>4%</td></tr><tr><td>PHP</td><td>3%</td></tr></tbody></table>	Language	Proportion (approx.)	JavaScript	18%	HTML/CSS	15%	SQL	12%	Python	10%	TypeScript	8%	Java	7%	Bash/Shell	6%	C#	5%	C++	4%	PHP	3%
Language	Proportion (approx.)																						
JavaScript	18%																						
HTML/CSS	15%																						
SQL	12%																						
Python	10%																						
TypeScript	8%																						
Java	7%																						
Bash/Shell	6%																						
C#	5%																						
C++	4%																						
PHP	3%																						

3. 많이 사용되는 프로그래밍 언어 그래프 그리기

■ 백분율 표기하기

- 명명인수 autopct 인수 → 백분율 표기 방식을 넣기
→ 표기 방식은 파이썬 문자열 포매팅 방식과 같음

In [24]:	<pre># 백분율 표기하기 size_by_languages.nlargest(10).plot.pie(figsize=(10, 10), autopct='%1.0f%%')</pre>																						
Out [24]:	 <table border="1"><thead><tr><th>Language</th><th>Percentage</th></tr></thead><tbody><tr><td>JavaScript</td><td>17%</td></tr><tr><td>HTML/CSS</td><td>14%</td></tr><tr><td>SQL</td><td>13%</td></tr><tr><td>Python</td><td>12%</td></tr><tr><td>TypeScript</td><td>9%</td></tr><tr><td>Java</td><td>9%</td></tr><tr><td>Bash/Shell</td><td>8%</td></tr><tr><td>C#</td><td>7%</td></tr><tr><td>C++</td><td>6%</td></tr><tr><td>PHP</td><td>5%</td></tr></tbody></table>	Language	Percentage	JavaScript	17%	HTML/CSS	14%	SQL	13%	Python	12%	TypeScript	9%	Java	9%	Bash/Shell	8%	C#	7%	C++	6%	PHP	5%
Language	Percentage																						
JavaScript	17%																						
HTML/CSS	14%																						
SQL	13%																						
Python	12%																						
TypeScript	9%																						
Java	9%																						
Bash/Shell	8%																						
C#	7%																						
C++	6%																						
PHP	5%																						

3. 원하는 연령대 데이터만 추출해 그래프 그리기

■ 25~34세 개발자들이 사용한 프로그래밍 언어 데이터를 추출하기

In [25]:	<pre># 25~34세 연령대가 사용한 프로그래밍 언어 정보 추출 languages_for_25_34 = revised_data[revised_data.Age == '25-34 years old'] ["LanguageHaveWorkedWith"].str.split(";").explode() # 데이터 확인 languages_for_25_34</pre>
Out [25]:	<pre>2 C# 2 C++ 2 HTML/CSS 2 JavaScript 2 Python ... 73267 JavaScript 73267 Lua 73267 PowerShell 73267 SQL 73267 TypeScript Name: LanguageHaveWorkedWith, Length: 137808, dtype: object</pre>

3. 원하는 연령대 데이터만 추출해 그래프 그리기

■ 상위 10개 프로그래밍 언어 파이 그래프 그리기

- 문제 해결

→ HTML/CSS, JavaScript, TypeScript, SQL, Python까지 배운다면, 소프트웨어 엔지니어로 취업하는 데 도움이 될 것으로 보임

In [26]:	<pre># 상위 10개 데이터 파이 그래프 그리기 languages_for_25_34.groupby(languages_for_25_34).size().nlargest(10). plot.pie(figsize=(10, 10), autopct='%1.0f%%')</pre>																						
Out [26]:	<p><AxesSubplot:ylabel='LanguageHaveWorkedWith'></p> <table border="1"><thead><tr><th>Language</th><th>Percentage</th></tr></thead><tbody><tr><td>JavaScript</td><td>18%</td></tr><tr><td>HTML/CSS</td><td>14%</td></tr><tr><td>SQL</td><td>13%</td></tr><tr><td>Python</td><td>12%</td></tr><tr><td>TypeScript</td><td>11%</td></tr><tr><td>Java</td><td>8%</td></tr><tr><td>Bash/Shell</td><td>7%</td></tr><tr><td>C#</td><td>7%</td></tr><tr><td>PHP</td><td>5%</td></tr><tr><td>C++</td><td>5%</td></tr></tbody></table>	Language	Percentage	JavaScript	18%	HTML/CSS	14%	SQL	13%	Python	12%	TypeScript	11%	Java	8%	Bash/Shell	7%	C#	7%	PHP	5%	C++	5%
Language	Percentage																						
JavaScript	18%																						
HTML/CSS	14%																						
SQL	13%																						
Python	12%																						
TypeScript	11%																						
Java	8%																						
Bash/Shell	7%																						
C#	7%																						
PHP	5%																						
C++	5%																						