In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
survey_raw_df = pd.read_csv(r'C:\Users\Faizal\Downloads\stack-overflow-developer-survey-
```

In [3]:

```
survey_raw_df
```

Out[3]:

	ResponseId	Q120	MainBranch	Age	Employment	RemoteWork	CodingActivities	EdLeve
0	1	I agree	None of these	18-24 years old	NaN	NaN	NaN	NaN
1	2	I agree	I am a developer by profession	25-34 years old	Employed, full-time	Remote	Hobby;Contribute to open-source projects;Boots...	Bachelor’s degree (B.A., B.S. B.Eng. etc.
2	3	I agree	I am a developer by profession	45-54 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Professional development or self-paced l...	Bachelor’s degree (B.A., B.S. B.Eng. etc.
3	4	I agree	I am a developer by profession	25-34 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor’s degree (B.A., B.S. B.Eng. etc.
4	5	I agree	I am a developer by profession	25-34 years old	Employed, full-time;Independent contractor, fr...	Remote	Hobby;Contribute to open-source projects;Profe...	Bachelor’s degree (B.A., B.S. B.Eng. etc.
...	...	...	...	...	...	...	...	..
89179	89180	I agree	I am a developer by profession	25-34 years old	Employed, full-time;Independent contractor, fr...	Remote	Hobby;Bootstrapping a business;Freelance/contr...	Associate degree (A.A., A.S. etc.
89180	89181	I agree	I am a developer by profession	18-24 years old	Student, full-time;Employed, part-time	Hybrid (some remote, some in-person)	School or academic work	Bachelor’s degree (B.A., B.S. B.Eng. etc.
89181	89182	I agree	I code primarily as a hobby	Prefer not to say	I prefer not to say	NaN	NaN	Something else
89182	89183	I agree	I am a developer by profession	Under 18 years old	Employed, part-time;Student, part-time	Hybrid (some remote, some in-person)	Hobby;School or academic work	Secondary school (e.g. American high school, G..
89183	89184	I agree	I am a developer by profession	35-44 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Professional development or self-paced l...	Bachelor’s degree (B.A., B.S.

B.Eng.
etc.

89184 rows × 84 columns

In [4]:
```
survey_raw_df.head()
```

Out[4]:

	ResponseId	Q120	MainBranch	Age	Employment	RemoteWork	CodingActivities	EdLevel	
0	1	I agree	None of these	18-24 years old	NaN	NaN	NaN	NaN	
1	2	I agree	I am a developer by profession	25-34 years old	Employed, full-time	Remote	Hobby;Contribute to open-source projects;Boots...	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Boc media;Coll
2	3	I agree	I am a developer by profession	45-54 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Professional development or self-paced l...	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Boc media;(
3	4	I agree	I am a developer by profession	25-34 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Colleag family m
4	5	I agree	I am a developer by profession	25-34 years old	Employed, full-time;Independent contractor, fr...	Remote	Hobby;Contribute to open-source projects;Profe...	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Boc media;Or

5 rows × 84 columns

In [5]:
```
# Display column names
print(survey_raw_df.columns)
```

```
Index(['ResponseId', 'Q120', 'MainBranch', 'Age', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'TechList', 'BuyNewTool', 'Country',
       'Currency', 'CompTotal', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysPersonal use',
       'OpSysProfessional use', 'OfficeStackAsyncHaveWorkedWith',
       'OfficeStackAsyncWantToWorkWith', 'OfficeStackSyncHaveWorkedWith',
       'OfficeStackSyncWantToWorkWith', 'AISearchHaveWorkedWith',
       'AISearchWantToWorkWith', 'AIDevHaveWorkedWith', 'AIDevWantToWorkWith',
       'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq', 'SOComm',
       'SOAI', 'AISelect', 'AISent', 'AIAcc', 'AIBen',
       'AIToolInterested in Using', 'AIToolCurrently Using',
       'AIToolNot interested in Using', 'AINextVery different',
       'AINextNeither different nor similar', 'AINextSomewhat similar',
       'AINextVery similar', 'AINextSomewhat different', 'TBranch', 'ICorPM',
       'WorkExp', 'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Knowledge_8',
       'Frequency_1', 'Frequency_2', 'Frequency_3', 'TimeSearching',
       'TimeAnswering', 'ProfessionalTech', 'Industry', 'SurveyLength',
       'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [6]:

```
# Display basic info about the dataframe
survey_raw_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89184 entries, 0 to 89183
Data columns (total 84 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   ResponseId                      89184 non-null  int64  
 1   Q120                            89184 non-null  object 
 2   MainBranch                      89184 non-null  object 
 3   Age                             89184 non-null  object 
 4   Employment                      87898 non-null  object 
 5   RemoteWork                      73810 non-null  object 
 6   CodingActivities                73764 non-null  object 
 7   EdLevel                         87973 non-null  object 
 8   LearnCode                       87663 non-null  object 
 9   LearnCodeOnline                 70084 non-null  object 
 10  LearnCodeCoursesCert            37076 non-null  object 
 11  YearsCode                       87435 non-null  object 
 12  YearsCodePro                    66136 non-null  object 
 13  DevType                         76872 non-null  object 
 14  OrgSize                         65043 non-null  object 
 15  PurchaseInfluence               64964 non-null  object 
 16  TechList                        60851 non-null  object 
 17  BuyNewTool                      83009 non-null  object 
 18  Country                         87973 non-null  object 
 19  Currency                        65334 non-null  object 
 20  CompTotal                       48225 non-null  float64
 21  LanguageHaveWorkedWith          87140 non-null  object 
 22  LanguageWantToWorkWith          80709 non-null  object 
 23  DatabaseHaveWorkedWith          73435 non-null  object 
 24  DatabaseWantToWorkWith          60911 non-null  object 
 25  PlatformHaveWorkedWith          63628 non-null  object 
 26  PlatformWantToWorkWith          51308 non-null  object 
 27  WebframeHaveWorkedWith          66938 non-null  object 
 28  WebframeWantToWorkWith          56741 non-null  object 
 29  MiscTechHaveWorkedWith          57019 non-null  object 
 30  MiscTechWantToWorkWith          46848 non-null  object 
 31  ToolsTechHaveWorkedWith         77884 non-null  object 
 32  ToolsTechWantToWorkWith         68315 non-null  object 
 33  NEWCollabToolsHaveWorkedWith    85864 non-null  object 
 34  NEWCollabToolsWantToWorkWith    76649 non-null  object 
 35  OpSysPersonal use               86557 non-null  object 
 36  OpSysProfessional use           78587 non-null  object 
 37  OfficeStackAsyncHaveWorkedWith  69090 non-null  object 
 38  OfficeStackAsyncWantToWorkWith  53743 non-null  object 
 39  OfficeStackSyncHaveWorkedWith   83439 non-null  object 
 40  OfficeStackSyncWantToWorkWith   69776 non-null  object 
 41  AISearchHaveWorkedWith          56328 non-null  object 
 42  AISearchWantToWorkWith          46150 non-null  object 
 43  AIDevHaveWorkedWith             25904 non-null  object 
 44  AIDevWantToWorkWith             19587 non-null  object 
 45  NEWSOSites                      87973 non-null  object 
 46  SOVisitFreq                     87140 non-null  object 
 47  SOAccount                       87852 non-null  object 
 48  SOPartFreq                      66061 non-null  object 
 49  SOComm                          87692 non-null  object 
 50  SOAI                                 47912 non-null  object 
 51  AISelect                             87973 non-null  object 
 52  AISent                               61501 non-null  object 
 53  AIAcc                                38594 non-null  object 
 54  AIBen                                61396 non-null  object 
 55  AIToolInterested in Using            32783 non-null  object 
 56  AIToolCurrently Using                36137 non-null  object 
 57  AIToolNot interested in Using        21069 non-null  object 
 58  AINextVery different                 12661 non-null  object 
 59  AINextNeither different nor similar  6599 non-null   object 
 60  AINextSomewhat similar               6238 non-null   object 
 61  AINextVery similar                   2621 non-null   object 
 62  AINextSomewhat different             23303 non-null  object 
 63  TBranch                              65768 non-null  object 
 64  ICorPM                               43668 non-null  object 
 65  WorkExp                              43579 non-null  float64
 66  Knowledge_1                          42535 non-null  object 
 67  Knowledge_2                          41670 non-null  object 
 68  Knowledge_3                          41798 non-null  object 
 69  Knowledge_4                          41684 non-null  object 
 70  Knowledge_5                          41527 non-null  object 
 71  Knowledge_6                          41520 non-null  object 
 72  Knowledge_7                          41467 non-null  object 
 73  Knowledge_8                          41404 non-null  object 
 74  Frequency_1                          41916 non-null  object 
 75  Frequency_2                          41925 non-null  object 
 76  Frequency_3                          41054 non-null  object 
 77  TimeSearching                        42778 non-null  object 
 78  TimeAnswering                        42629 non-null  object 
 79  ProfessionalTech                     41783 non-null  object 
 80  Industry                             36774 non-null  object 
 81  SurveyLength                         86485 non-null  object 
 82  SurveyEase                           86554 non-null  object 
 83  ConvertedCompYearly                  48019 non-null  float64
dtypes: float64(3), int64(1), object(80)
memory usage: 57.2+ MB
```

In [7]:

```
# Display summary statistics
survey_raw_df.describe()
```

Out[7]:

	ResponseId	CompTotal	WorkExp	ConvertedCompYearly
count	89184.000000	4.822500e+04	43579.000000	4.801900e+04
mean	44592.500000	1.036807e+42	11.405126	1.031101e+05
std	25745.347541	2.276847e+44	9.051989	6.814188e+05
min	1.000000	0.000000e+00	0.000000	1.000000e+00
25%	22296.750000	6.300000e+04	5.000000	4.390700e+04
50%	44592.500000	1.150000e+05	9.000000	7.496300e+04
75%	66888.250000	2.300000e+05	16.000000	1.216410e+05
max	89184.000000	5.000000e+46	50.000000	7.435143e+07

In [8]:

```
### Loading schema dataframe
schema_fname = r'C:\Users\Faizal\Downloads\stack-overflow-developer-survey-2023\survey_r
schema_raw = pd.read_csv(schema_fname, index_col='Columns').QuestionText
```

In [9]:

```
schema_raw
```

Out[9]:

```
Columns
Respondent              Randomized respondent ID number (not in order ...
Q120                            Agreeing Terms & Conditions to fill the survey
MainBranch                  Which of the following options best describes ...
Age                                                          What is your age? *
Employment                  Which of the following best describes your cur...
                                                          ...
ProfessionalTech            Since how many years you have been into profes...
Industry                        What industry is the company you work for in?
SurveyLength                How do you feel about the length of the survey...
SurveyEase                  How easy or difficult was this survey to compl...
ConvertedCompYearly         What is your converted current total annual co...
Name: QuestionText, Length: 84, dtype: object
```

In [10]:

```python
schema_raw['YearsCodePro']
```

Out[10]:

```
'NOT including education, how many years have you coded professionally (as a part of you
r work)?'
```

Data Preparation & Cleaning

While the survey responses contain a wealth of information, we'll limit our analysis to the following areas:

Demographics of the survey respondents and the global programming community Distribution of programming skills, experience, and preferences Employment-related information, preferences, and opinions Let's select a subset of columns with the relevant data for our analysis.

In [11]:

```python
selected_columns = [
    # Demographics
    'Country',
    'Age',
    'EdLevel',
    # Programming experience
    'YearsCode',
    'YearsCodePro',
    'LanguageHaveWorkedWith',
    'LanguageWantToWorkWith',
    'MiscTechWantToWorkWith',
    'ToolsTechHaveWorkedWith',
    'ToolsTechWantToWorkWith',
    'DatabaseHaveWorkedWith',
    'DatabaseWantToWorkWith',
    'PlatformHaveWorkedWith',
    'PlatformWantToWorkWith',
    'WebframeHaveWorkedWith',
    'WebframeWantToWorkWith',
    # Employment
    'Employment',
    'DevType',
    'WorkExp',
    'ProfessionalTech',
    'Industry',
    # Remote Work and Work Environment
    'RemoteWork',
    'OrgSize',
    # Compensation
    'CompTotal',
    'ConvertedCompYearly',
    # Community Engagement and Learning
    'SOVisitFreq',
    'SOPartFreq',
    'LearnCode',
    'LearnCodeOnline',
    # AI and Tools
    'AISearchHaveWorkedWith',
    'AISearchWantToWorkWith'
]
```

In [12]:

```
len(selected_columns)
```

Out[12]:

```
31
```

Let's extract a copy of the data from these columns into a new data frame survey_df. We can continue to modify further without affecting the original data frame.

In [13]:

```
survey_df = survey_raw_df[selected_columns].copy()
```

In [14]:

```
schema_df = schema_raw[selected_columns].copy()
```

In [15]:

```
survey_df.shape
```

Out[15]:

```
(89184, 31)
```

In [16]:

```
survey_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89184 entries, 0 to 89183
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Country                 87973 non-null  object 
 1   Age                     89184 non-null  object 
 2   EdLevel                 87973 non-null  object 
 3   YearsCode               87435 non-null  object 
 4   YearsCodePro            66136 non-null  object 
 5   LanguageHaveWorkedWith  87140 non-null  object 
 6   LanguageWantToWorkWith  80709 non-null  object 
 7   MiscTechWantToWorkWith  46848 non-null  object 
 8   ToolsTechHaveWorkedWith 77884 non-null  object 
 9   ToolsTechWantToWorkWith 68315 non-null  object 
 10  DatabaseHaveWorkedWith  73435 non-null  object 
 11  DatabaseWantToWorkWith  60911 non-null  object 
 12  PlatformHaveWorkedWith  63628 non-null  object 
 13  PlatformWantToWorkWith  51308 non-null  object 
 14  WebframeHaveWorkedWith  66938 non-null  object 
 15  WebframeWantToWorkWith  56741 non-null  object 
 16  Employment              87898 non-null  object 
 17  DevType                 76872 non-null  object 
 18  WorkExp                 43579 non-null  float64
 19  ProfessionalTech        41783 non-null  object 
 20  Industry                36774 non-null  object 
 21  RemoteWork              73810 non-null  object 
 22  OrgSize                 65043 non-null  object 
 23  CompTotal               48225 non-null  float64
 24  ConvertedCompYearly     48019 non-null  float64
 25  SOVisitFreq             87140 non-null  object 
 26  SOPartFreq              66061 non-null  object 
 27  LearnCode               87663 non-null  object 
 28  LearnCodeOnline         70084 non-null  object 
 29  AISearchHaveWorkedWith  56328 non-null  object 
 30  AISearchWantToWorkWith  46150 non-null  object 
dtypes: float64(3), object(28)
memory usage: 21.1+ MB
```

Datatype conversion within columns

In [17]:
```
survey_df['YearsCode'] = pd.to_numeric(survey_df.YearsCode, errors='coerce')
survey_df['YearsCodePro'] = pd.to_numeric(survey_df.YearsCodePro, errors='coerce')
```

In [18]:
```
survey_df.describe()
```

Out[18]:

	YearsCode	YearsCodePro	WorkExp	CompTotal	ConvertedCompYearly
count	86114.000000	64217.000000	43579.000000	4.822500e+04	4.801900e+04
mean	13.972049	11.508354	11.405126	1.036807e+42	1.031101e+05
std	10.259548	9.070346	9.051989	2.276847e+44	6.814188e+05
min	1.000000	1.000000	0.000000	0.000000e+00	1.000000e+00
25%	6.000000	5.000000	5.000000	6.300000e+04	4.390700e+04
50%	11.000000	9.000000	9.000000	1.150000e+05	7.496300e+04
75%	20.000000	16.000000	16.000000	2.300000e+05	1.216410e+05
max	50.000000	50.000000	50.000000	5.000000e+46	7.435143e+07

Dropping some data within the dataset

Replacing the NA values within column

In [19]:
```
survey_df['EdLevel'].fillna('Other', inplace=True)
survey_df['Employment'].fillna('Not Employed', inplace=True)
```

In [20]:
```
survey_df.sample(10)
```

Out[20]:

	Country	Age	EdLevel	YearsCode	YearsCodePro	LanguageHaveWorke
86627	Greece	18-24 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	10.0	NaN	Assembly;Bash/Shell (all shells);C;C++;Ja
83996	India	25-34 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	10.0	2.0	
62531	Canada	25-34 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	15.0	11.0	Bash/Shell (all shells);Groovy;HTML/CSS;J
52000	Bangladesh	25-34 years old	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)	10.0	NaN	HTML/CSS;JavaScript;Python;SQL;Type
52911	Canada	35-44 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	27.0	17.0	Crysta
53523	Pakistan	18-24 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	3.0	NaN	Assembly;C++;HTML/CSS;JavaScript;MATLAB;F
18141	United Kingdom of Great	25-34 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	5.0	5.0	Bash/Shell (all shells);Go;HTML/CSS;JavaS

	Britain and Northern I...					
27118	El Salvador	25-34 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	13.0	7.0	Bash/Shell (all shells);HTML/CSS;Java;Jav
34780	United States of America	Under 18 years old	Primary/elementary school	3.0	NaN	Assembly;Bash/Shell (all shells);C;Elixir;H
2441	United States of America	35-44 years old	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	11.0	9.0	Bash/Shell (all shells);HTML/CSS;JavaScri

10 rows × 31 columns

Exploratory Data Analysis and Visualization

In [21]:

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

```
C:\Users\Faizal\anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarning: A NumPy
version >=1.18.5 and <1.25.0 is required for this version of SciPy (detected version 1.2
6.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Country

Let's look at the number of countries from which there are responses in the survey and plot the ten countries with the highest number of responses.

In [22]:

```
schema_df.Country
```

Out[22]:

```
'Where do you live? '
```

In [23]:

```
survey_df.Country.nunique()
```

Out[23]:

```
185
```

We can identify the countries with the highest number of respondents using the value_counts method.

In [24]:

```
top_countries = survey_df.Country.value_counts().head(15)
top_countries
```

Out[24]:

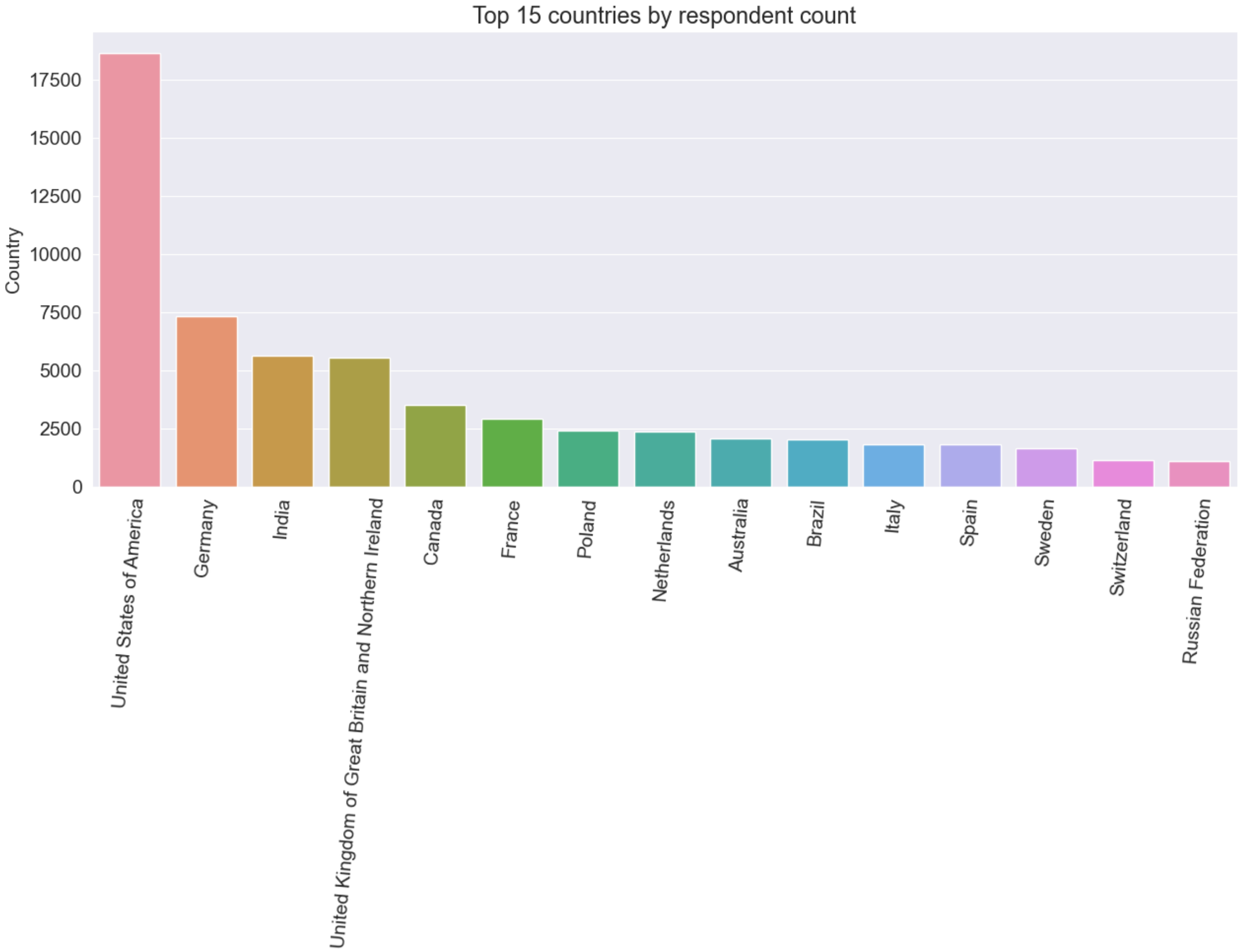
```
United States of America                                 18647
Germany                                                   7328
```

```
India                                                    5625
United Kingdom of Great Britain and Northern Ireland     5552
Canada                                                   3507
France                                                   2933
Poland                                                   2435
Netherlands                                              2383
Australia                                                2078
Brazil                                                   2042
Italy                                                    1835
Spain                                                    1834
Sweden                                                   1641
Switzerland                                              1149
Russian Federation                                       1094
Name: Country, dtype: int64
```

The survey captured responses from a diverse set of countries. The top countries by the number of respondents are the United States, India, Germany, the United Kingdom, and Canada. These countries represent the bulk of the global developer workforce, highlighting where most developers are located.

In [49]:

```python
plt.figure(figsize=(15,6))
plt.xticks(rotation=85)
plt.title('Top 15 countries by respondent count')
sns.barplot(x=top_countries.index, y=top_countries);
```

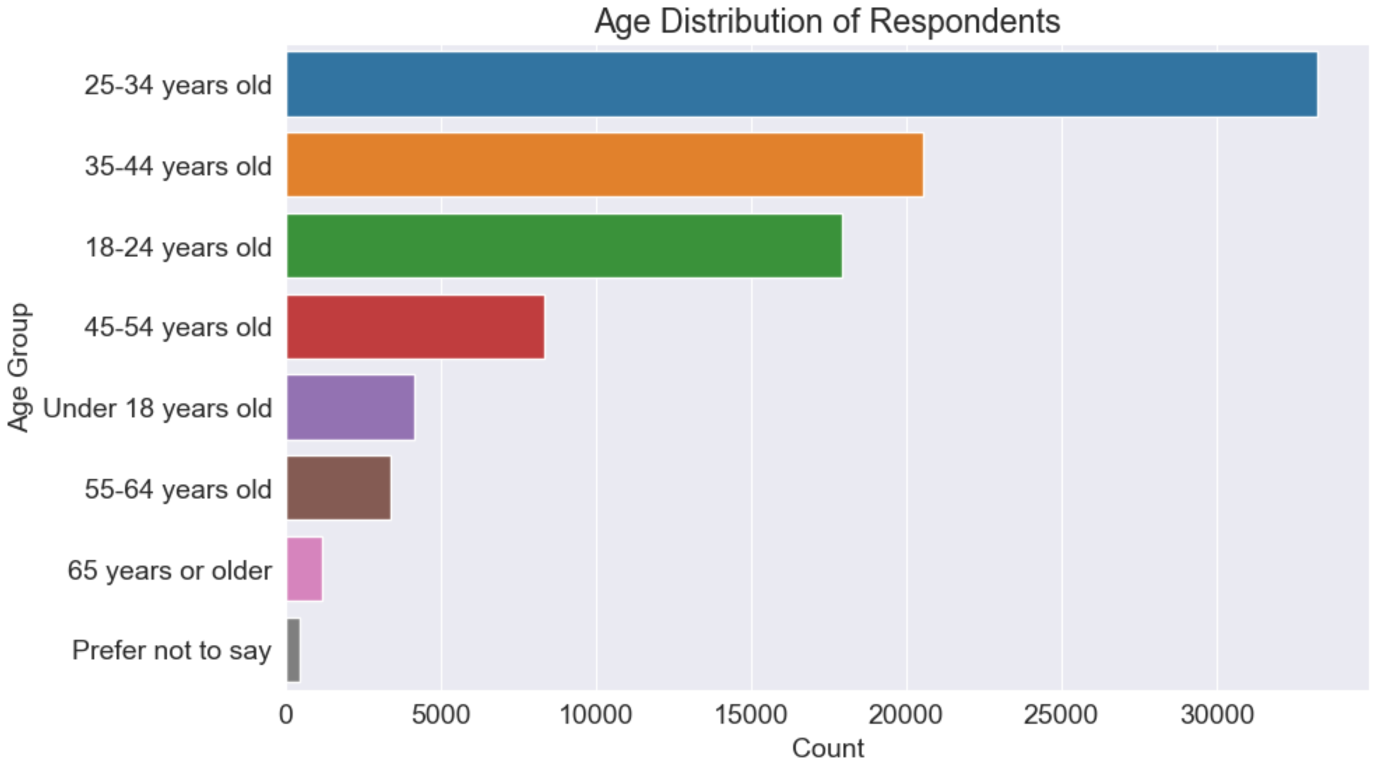


Age

Respondents' ages range widely, but a significant majority fall within the 25-34 age group. This suggests that the developer community is predominantly composed of young professionals, which aligns with the typical career stage for many in this field.

The distribution of respondents' age is another crucial factor to look at. We can use countplot to visualize it.

```
In [26]:
# Drop NA values
survey_df['Age'] = survey_df['Age'].dropna()

# Age distribution
plt.figure(figsize=(10, 6))
plt.title('Age Distribution of Respondents')
sns.countplot(y=survey_df['Age'], order=survey_df['Age'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Age Group')
plt.show()
```

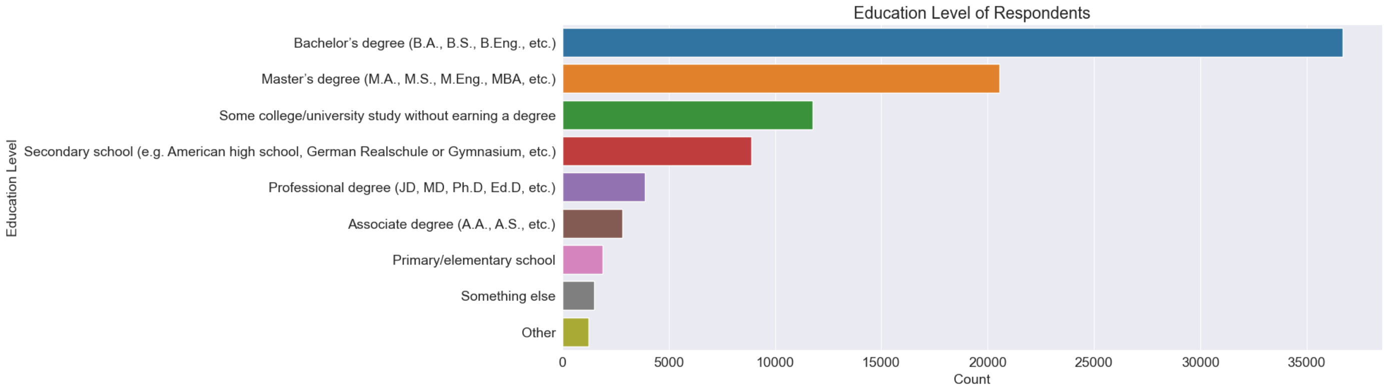


Education Level

Most respondents have attained at least a bachelor's degree, with a substantial number holding a master's degree or higher. This high level of education reflects the technical nature of the profession and the necessity for formal education in computer science or related fields.

Formal education in computer science is often considered an essential requirement for becoming a programmer. However, there are many free resources & tutorials available online to learn programming. Let's compare the education levels of respondents to gain some insight into this. We'll use a horizontal bar plot here.

```
In [27]:
plt.figure(figsize=(15,6))
plt.title('Education Level of Respondents')
sns.countplot(y=survey_df['EdLevel'], order=survey_df['EdLevel'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Education Level')
plt.show()
```



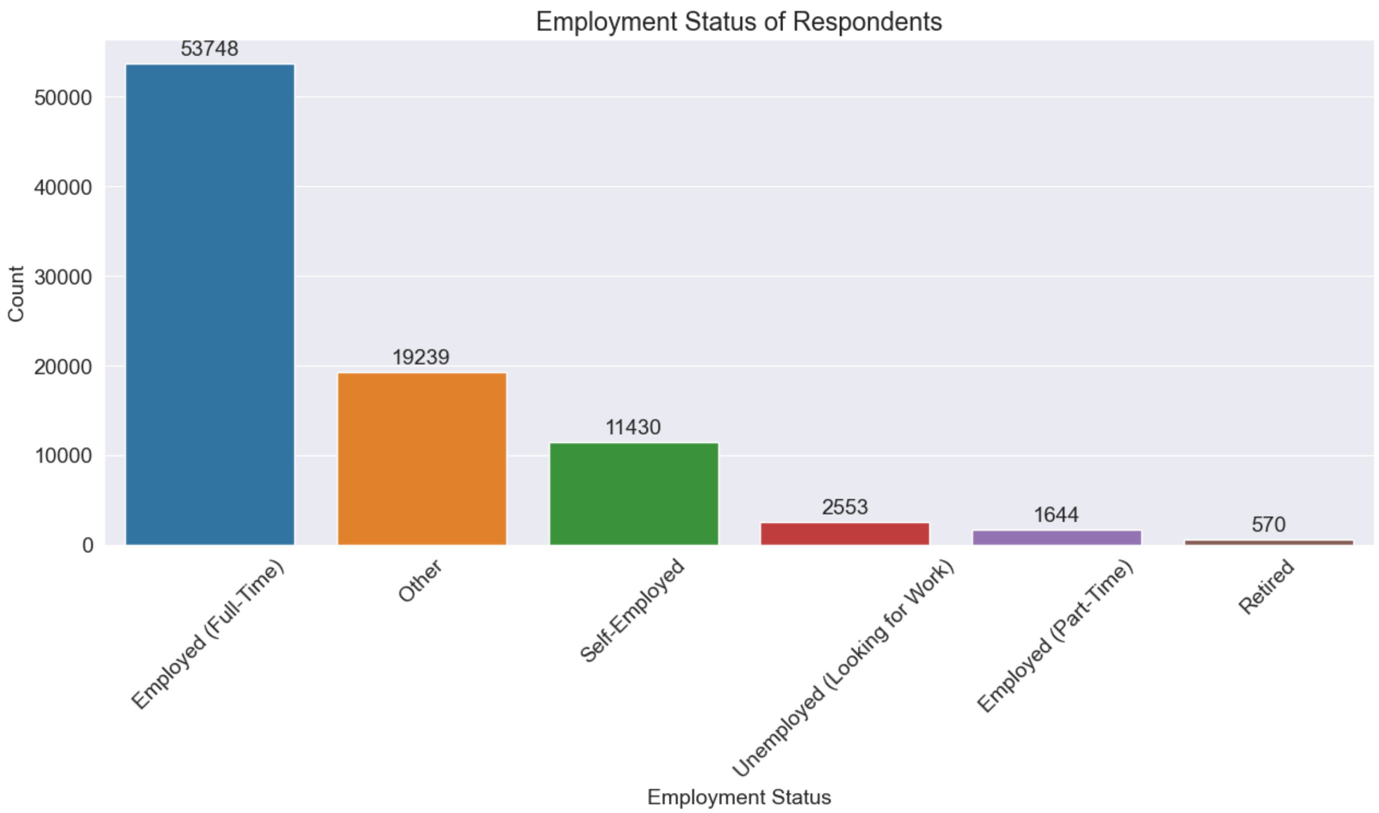
Employment

Employment Status: A majority of respondents are employed full-time, with a notable portion working as independent contractors or freelancers. There is also a significant number of respondents looking for work, indicating a dynamic job market.

Freelancing or contract work is a common choice among programmers, so it would be interesting to compare the breakdown between full-time, part-time, and freelance work. Let's visualize the data from the Employment column.

In [28]:
```
# Define mapping to broader categories
employment_mapping = {
    'Employed, full-time': 'Employed (Full-Time)',
    'Employed, part-time': 'Employed (Part-Time)',
    'Employed, full-time;Independent contractor, freelancer, or self-employed': 'Self-Em
    'Independent contractor, freelancer, or self-employed': 'Self-Employed',
    'Not employed, but looking for work': 'Unemployed (Looking for Work)',
    'Retired': 'Retired',
    'Prefer not to say': 'Prefer Not to Say'
}

# Map detailed statuses to broader categories
survey_df['EmploymentCategory'] = survey_df['Employment'].map(employment_mapping).fillna

# Count occurrences of each employment category
employment_counts = survey_df['EmploymentCategory'].value_counts()
```

In [29]:
```
# Plot
plt.figure(figsize=(15, 6))
bar_plot = sns.barplot(x=employment_counts.index, y=employment_counts.values)
plt.xlabel('Employment Status')
plt.ylabel('Count')
plt.title('Employment Status of Respondents')
plt.xticks(rotation=45)

# Add data labels
for p in bar_plot.patches:
    bar_plot.annotate(format(int(p.get_height())),
                      (p.get_x() + p.get_width() / 2., p.get_height()),
                      ha = 'center', va = 'center',
                      xytext = (0, 9),
                      textcoords = 'offset points')

plt.show()
```

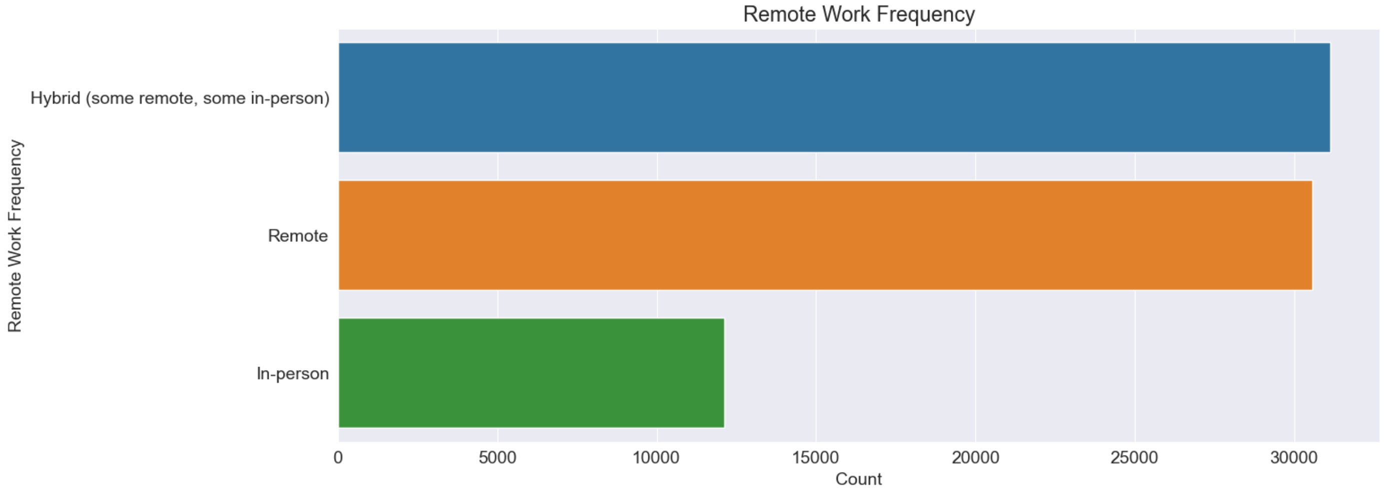


RemoteWork

A significant portion of respondents work remotely, either fully or partially. The trend towards remote work has been accelerated by the COVID-19 pandemic and continues to be a preferred arrangement for many developers. Remote work offers flexibility and access to global opportunities, contributing to job satisfaction and productivity.

In [30]:

```
plt.figure(figsize=(15,6))
plt.title('Remote Work Frequency')
sns.countplot(y=survey_df['RemoteWork'], order=survey_df['RemoteWork'].value_counts().in
plt.xlabel('Count')
plt.ylabel('Remote Work Frequency')
plt.show()
```

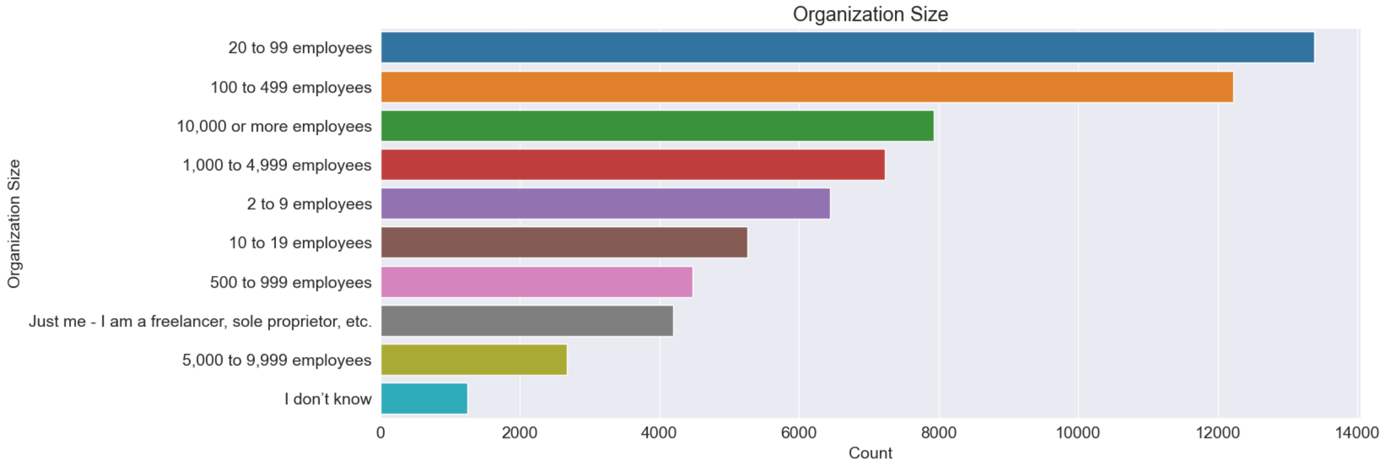


Organization Size

Respondents come from organizations of all sizes, from small startups to large enterprises. This diversity highlights the varied environments in which developers operate, each with unique challenges and opportunities.

In [31]:

```
plt.figure(figsize=(15,6))
plt.title('Organization Size')
sns.countplot(y=survey_df['OrgSize'], order=survey_df['OrgSize'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Organization Size')
plt.show()
```

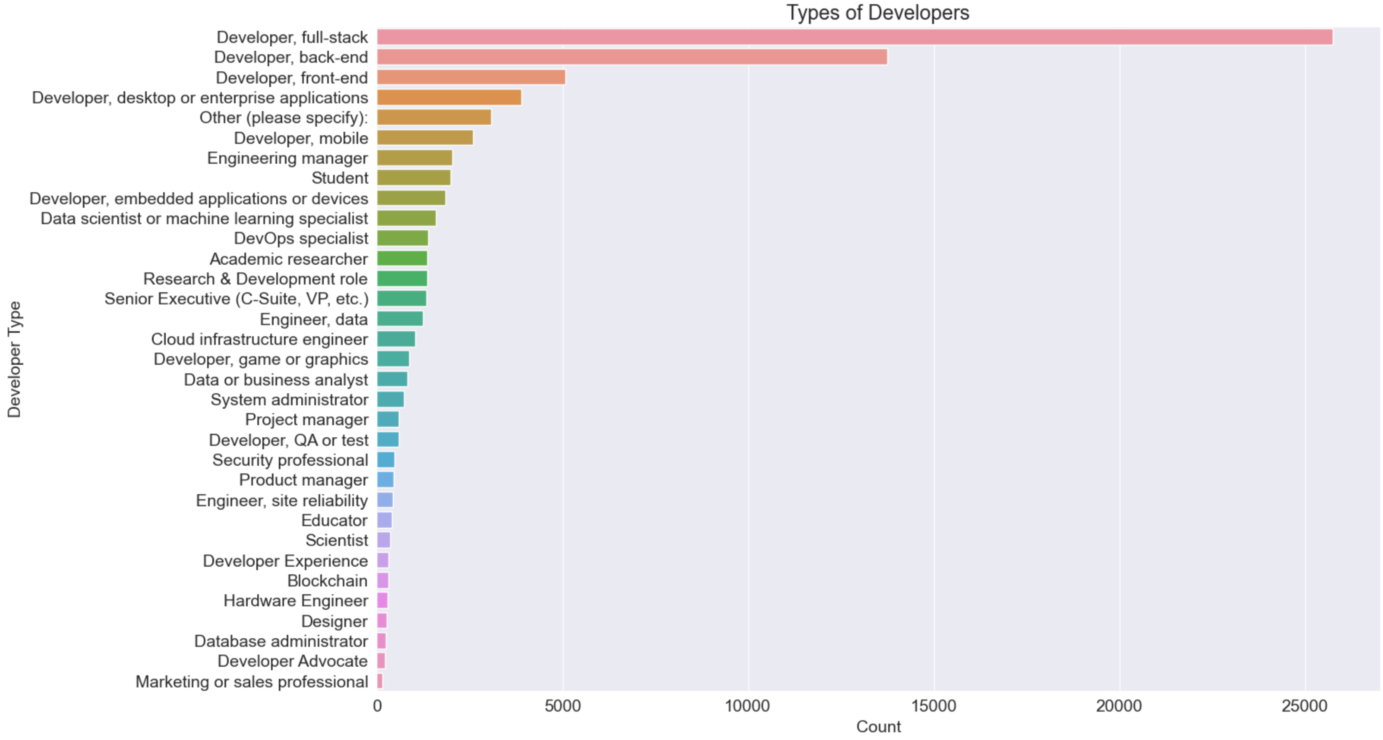


Developer Types

The survey includes various developer roles, with web developers and backend developers being the most common. Other prominent roles include full-stack developers, mobile developers, and DevOps specialists. This diversity in roles shows the multifaceted nature of the software development industry.

In [32]:

```
plt.figure(figsize=(15,10))
plt.title('Types of Developers')
sns.countplot(y=survey_df['DevType'], order=survey_df['DevType'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Developer Type')
plt.show()
```



The DevType field contains information about the roles held by respondents. Since the question allows multiple answers, the column contains lists of values separated by a semi-colon ;, making it a bit harder to analyze directly.

```
In [33]: schema_df.DevType
```

```
Out[33]: 'Which of the following describes your current job, the one you do most of the time? Ple
ase select only one.'
```

```
In [34]: survey_df.DevType.value_counts()
```

```
Out[34]:
Developer, full-stack                            25735
Developer, back-end                              13745
Developer, front-end                              5071
Developer, desktop or enterprise applications     3904
Other (please specify):                           3080
Developer, mobile                                 2597
Engineering manager                               2033
Student                                           1996
Developer, embedded applications or devices       1845
Data scientist or machine learning specialist     1588
DevOps specialist                                 1387
Academic researcher                               1354
Research & Development role                       1353
Senior Executive (C-Suite, VP, etc.)              1332
Engineer, data                                    1248
Cloud infrastructure engineer                     1036
Developer, game or graphics                        866
Data or business analyst                           837
System administrator                               743
Project manager                                    589
Developer, QA or test                              586
Security professional                              474
Product manager                                    446
Engineer, site reliability                         427
Educator                                           415
Scientist                                          351
Developer Experience                               326
Blockchain                                         323
Hardware Engineer                                  286
```

```
Designer                                          281
Database administrator                            257
Developer Advocate                                212
Marketing or sales professional                   149
Name: DevType, dtype: int64
```

In [35]:

```python
def split_multicolumn(col_series):
    result_df = col_series.to_frame()
    options = []
    # Iterate over the column
    for idx, value  in col_series[col_series.notnull()].iteritems():
        # Break each value into list of options
        for option in value.split(';'):
            # Add the option as a column to result
            if not option in result_df.columns:
                options.append(option)
                result_df[option] = False
            # Mark the value in the option column as True
            result_df.at[idx, option] = True
    return result_df[options]
```

In [36]:

```python
dev_type_df = split_multicolumn(survey_df.DevType)
```

In [37]:

```python
dev_type_df
```

Out[37]:

	Senior Executive (C-Suite, VP, etc.)	Developer, back-end	Developer, front-end	Developer, full-stack	System administrator	Developer, desktop or enterprise applications	Developer, QA or test	Designer	Data scientist or machine learning specialist
0	False	False	False	False	False	False	False	False	False
1	True	False	False	False	False	False	False	False	False
2	False	True	False	False	False	False	False	False	False
3	False	False	True	False	False	False	False	False	False
4	False	False	False	True	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
89179	False	False	True	False	False	False	False	False	False
89180	False	True	False	False	False	False	False	False	False
89181	False	False	False	False	False	False	False	False	False
89182	False	False	False	False	True	False	False	False	False
89183	False	False	False	True	False	False	False	False	False

89184 rows × 33 columns

In [38]:

```python
dev_type_totals = dev_type_df.sum().sort_values(ascending=False)
dev_type_totals
```

Out[38]:

```
Developer, full-stack                           25735
Developer, back-end                             13745
Developer, front-end                             5071
Developer, desktop or enterprise applications    3904
Other (please specify):                          3080
Developer, mobile                                2597
Engineering manager                              2033
```

```
Student                                          1996
Developer, embedded applications or devices      1845
Data scientist or machine learning specialist    1588
DevOps specialist                                1387
Academic researcher                              1354
Research & Development role                      1353
Senior Executive (C-Suite, VP, etc.)             1332
Engineer, data                                   1248
Cloud infrastructure engineer                    1036
Developer, game or graphics                       866
Data or business analyst                          837
System administrator                              743
Project manager                                   589
Developer, QA or test                             586
Security professional                             474
Product manager                                   446
Engineer, site reliability                        427
Educator                                          415
Scientist                                         351
Developer Experience                              326
Blockchain                                        323
Hardware Engineer                                 286
Designer                                          281
Database administrator                            257
Developer Advocate                                212
Marketing or sales professional                   149
dtype: int64
```

Languages used

The most commonly used languages include JavaScript, HTML/CSS, SQL, Python, and TypeScript. JavaScript remains the dominant language, essential for web development, followed closely by HTML/CSS and SQL. The popularity of Python and TypeScript highlights their growing importance in both web and general-purpose development.

In [39]:

```
survey_df.LanguageHaveWorkedWith
```

Out[39]:

```
0                                                      NaN
1                                HTML/CSS;JavaScript;Python
2                                Bash/Shell (all shells);Go
3         Bash/Shell (all shells);HTML/CSS;JavaScript;PH...
4                            HTML/CSS;JavaScript;TypeScript
                                ...                        
89179               HTML/CSS;Java;JavaScript;SQL;TypeScript
89180                                  Dart;Java;Python;SQL
89181     Assembly;Bash/Shell (all shells);C;C#;Python;R...
89182     Bash/Shell (all shells);C#;HTML/CSS;Java;JavaS...
89183                           C#;Go;JavaScript;SQL;TypeScript
Name: LanguageHaveWorkedWith, Length: 89184, dtype: object
```

In [40]:

```
## Splitting the columns into individual programming languages
languages_worked_df = split_multicolumn(survey_df.LanguageHaveWorkedWith)
```

In [41]:

```
languages_worked_df
```

Out[41]:

	HTML/CSS	JavaScript	Python	Bash/Shell (all shells)	Go	PHP	Ruby	SQL	TypeScript	Ada	...	GDScript	Crys
0	False	False	False	False	False	False	False	False	False	False	...	False	Fa
1	True	True	True	False	False	False	False	False	False	False	...	False	Fa

2	False	False	False	True	True	False	False	False	False	False	...	False	Fa
3	True	True	False	True	False	True	True	True	True	False	...	False	Fa
4	True	True	False	False	False	False	False	False	True	False	...	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	...	
89179	True	True	False	False	False	False	False	True	True	False	...	False	Fa
89180	False	False	True	False	False	False	False	True	False	False	...	False	Fa
89181	False	False	True	True	False	False	False	True	False	False	...	False	Fa
89182	True	True	True	True	False	False	False	True	False	False	...	False	Fa
89183	False	True	False	False	True	False	False	True	True	False	...	False	Fa

89184 rows × 51 columns

In [50]:

```
## converting values into percenatges
languages_worked_percentages = languages_worked_df.mean().sort_values(ascending=False) *
languages_worked_percentages
```

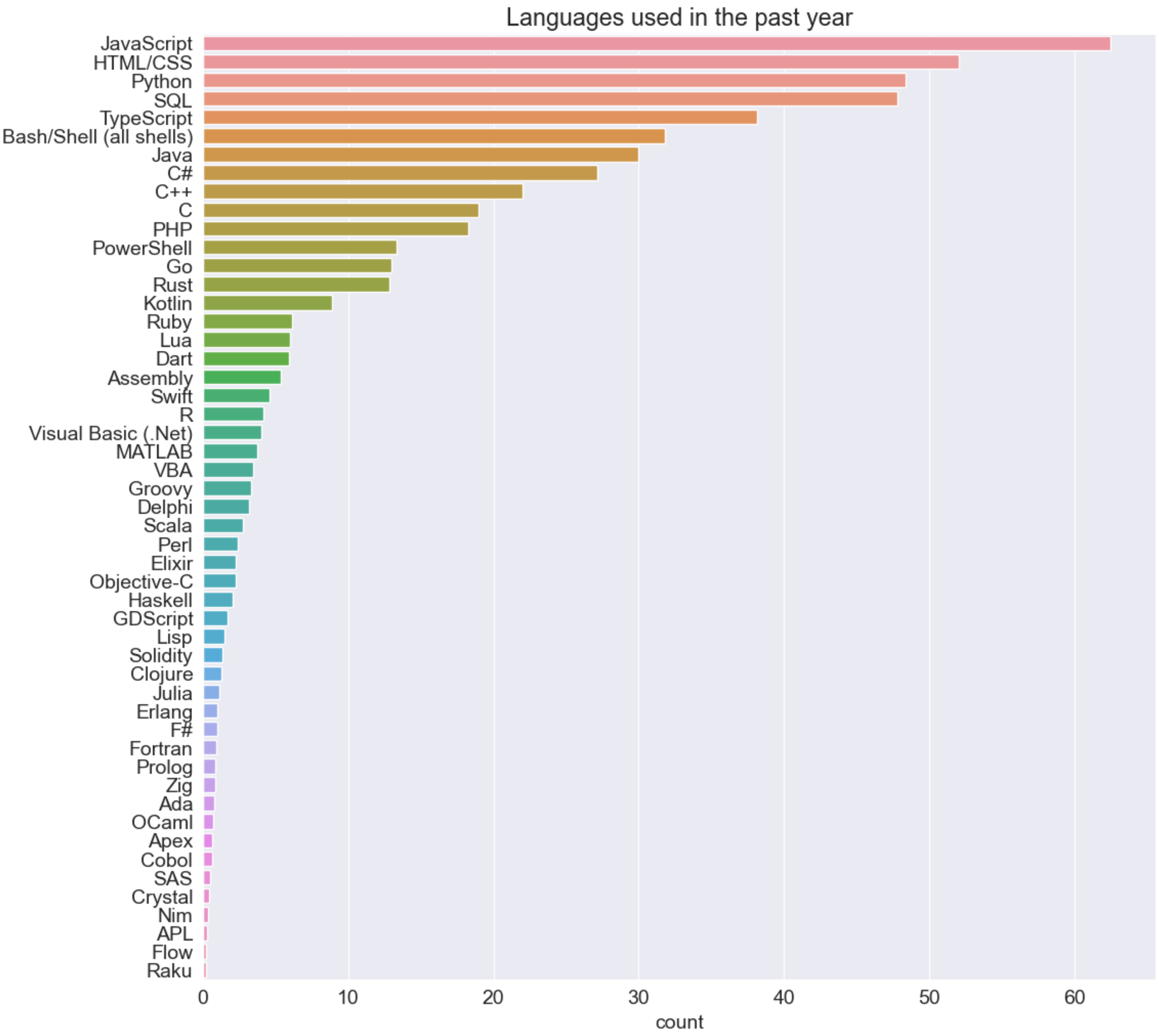
Out[50]:

```
JavaScript                 62.467483
HTML/CSS                   52.022784
Python                     48.392088
SQL                        47.792205
TypeScript                 38.169403
Bash/Shell (all shells)    31.789334
Java                       30.002018
C#                         27.127063
C++                        22.015160
C                          18.994438
PHP                        18.247668
PowerShell                 13.345443
Go                         12.997847
Rust                       12.812836
Kotlin                      8.897336
Ruby                        6.115447
Lua                         5.983136
Dart                        5.912496
Assembly                    5.329431
Swift                       4.565841
R                           4.150969
Visual Basic (.Net)         4.000718
MATLAB                      3.743945
VBA                         3.483809
Groovy                      3.336921
Delphi                      3.174336
Scala                       2.715734
Perl                        2.411868
Elixir                      2.273950
Objective-C                 2.263859
Haskell                     2.050816
GDScript                    1.676310
Lisp                        1.504754
Solidity                    1.309652
Clojure                     1.239011
Julia                       1.132490
Erlang                      0.973269
F#                          0.951964
Fortran                     0.934024
Prolog                      0.870111
Zig                         0.817411
Ada                         0.759105
```

```
OCaml                        0.688464
Apex                         0.649220
Cobol                        0.645856
SAS                          0.478785
Crystal                      0.436177
Nim                          0.371143
APL                          0.252287
Flow                         0.239953
Raku                         0.174919
dtype: float64
```

In [43]:

```
plt.figure(figsize=(12, 12))
sns.barplot(x=languages_worked_percentages, y=languages_worked_percentages.index)
plt.title("Languages used in the past year");
plt.xlabel('count');
```



Perhaps unsurprisingly, Javascript & HTML/CSS comes out at the top as web development is one of today's most sought skills. It also happens to be one of the easiest to get started. SQL is necessary for working with relational databases, so it's no surprise that most programmers work with SQL regularly. Python seems to be the popular choice for other forms of development, beating out Java, which was the industry standard for server & application development for over two decades.

Q: Which languages are the most people interested to learn over the next year?

For this, we can use the LanguageDesireNextYear column, with similar processing as the previous one.

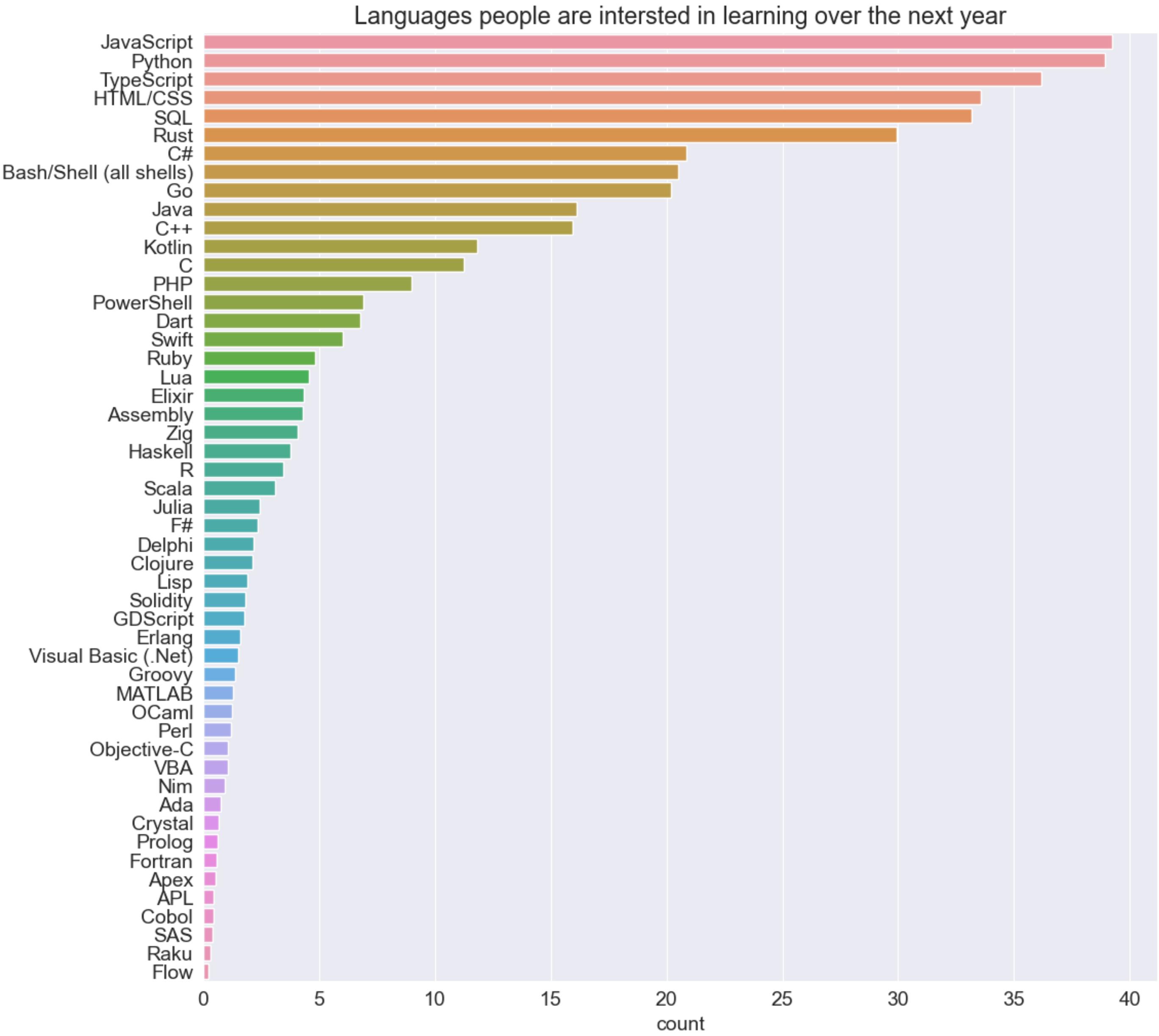
In [44]:
```
languages_interested_df = split_multicolumn(survey_df.LanguageWantToWorkWith)
languages_interested_percentages = languages_interested_df.mean().sort_values(ascending=
languages_interested_percentages
```

Out[44]:
```
JavaScript                 39.229010
Python                     38.925144
TypeScript                 36.167922
HTML/CSS                   33.557589
SQL                        33.187567
Rust                       29.949318
C#                         20.847910
Bash/Shell (all shells)    20.495829
Go                         20.175144
Java                       16.128454
C++                        15.956898
Kotlin                     11.838446
C                          11.245291
PHP                         8.985917
PowerShell                  6.886886
Dart                        6.790456
Swift                       6.040321
Ruby                        4.813644
Lua                         4.577054
Elixir                      4.350556
Assembly                    4.294492
Zig                         4.089299
Haskell                     3.767492
R                           3.440079
Scala                       3.105938
Julia                       2.448870
F#                          2.355804
Delphi                      2.184248
Clojure                     2.141640
Lisp                        1.911778
Solidity                    1.790680
GDScript                    1.754799
Erlang                      1.587729
Visual Basic (.Net)         1.513724
Groovy                      1.369080
MATLAB                      1.277135
OCaml                       1.214343
Perl                        1.208737
Objective-C                 1.069699
VBA                         1.065213
Nim                         0.909356
Ada                         0.740043
Crystal                     0.671645
Prolog                      0.625673
Fortran                     0.569609
Apex                        0.539334
APL                         0.456360
Cobol                       0.442905
SAS                         0.398053
Raku                        0.312836
Flow                        0.202951
dtype: float64
```

Languages to Learn:

Many respondents express interest in learning new languages, with Python and TypeScript being the most desired. This reflects the ongoing evolution of the technology landscape and the need for developers to continuously update their skill sets.

In [45]:
```
plt.figure(figsize=(12, 12))
sns.barplot(x=languages_interested_percentages, y=languages_interested_percentages.index
plt.title("Languages people are intersted in learning over the next year");
plt.xlabel('count');
```



Once again, it's not surprising that JavaScript and Python is the language most people are interested in learning - since it is an easy-to-learn general-purpose programming language well suited for a variety of domains: application development, numerical computing, data analysis, machine learning, big data, cloud automation, web scraping, scripting, etc. We're using Python for this very analysis, so we're in good company!

Q: Which are the most loved languages, i.e., a high percentage of people who have used the language

want to continue learning & using it over the next year?

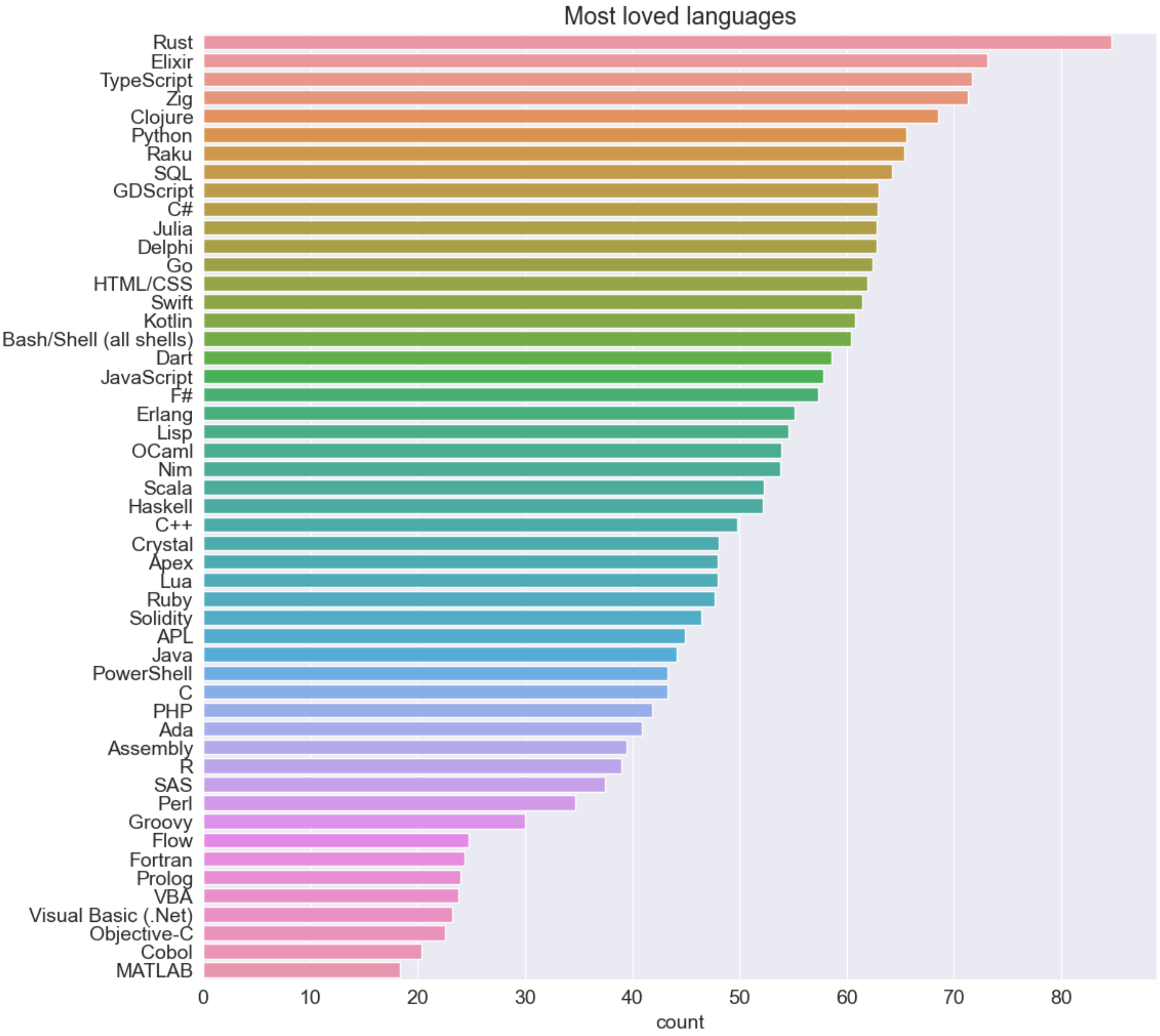
While this question may seem tricky at first, it's straightforward to solve using Pandas array operations. Here's what we can do:

Create a new data frame languages_loved_df that contains a True value for a language only if the corresponding values in languages_worked_df and languages_interested_df are both True Take the column-wise sum of languages_loved_df and divide it by the column-wise sum of languages_worked_df to get the percentage of respondents who "love" the language Sort the results in decreasing order and plot a horizontal bar graph

In [46]:
```
languages_loved_df = languages_worked_df & languages_interested_df
```

In [47]:
```
languages_loved_percentages = (languages_loved_df.sum() * 100/ languages_worked_df.sum()
```

In [48]:
```
plt.figure(figsize=(12, 12))
sns.barplot(x=languages_loved_percentages, y=languages_loved_percentages.index)
plt.title("Most loved languages");
plt.xlabel('count');
```



Rust has been StackOverflow's most-loved language for many years in a row. The second most-loved language is Elixir, and Typescript at number 3 which is a popular alternative to JavaScript for web development.

Python features at number 6, despite already being one of the most widely-used languages in the world. Python has a solid foundation, is easy to learn & use, has a large ecosystem of domain-specific libraries, and a massive worldwide community.

Summary:

The Stack Overflow Developer Survey 2023 offers valuable insights into the developer community. Key findings include:

A young, well-educated workforce predominantly located in a few key countries. A mix of experienced professionals and new entrants, with a strong interest in emerging languages and technologies. A shift towards remote work and diverse organizational environments. Active engagement in community learning and continuous skill development. These insights can guide educational institutions, employers, and policymakers in making informed decisions to support and nurture the developer community.