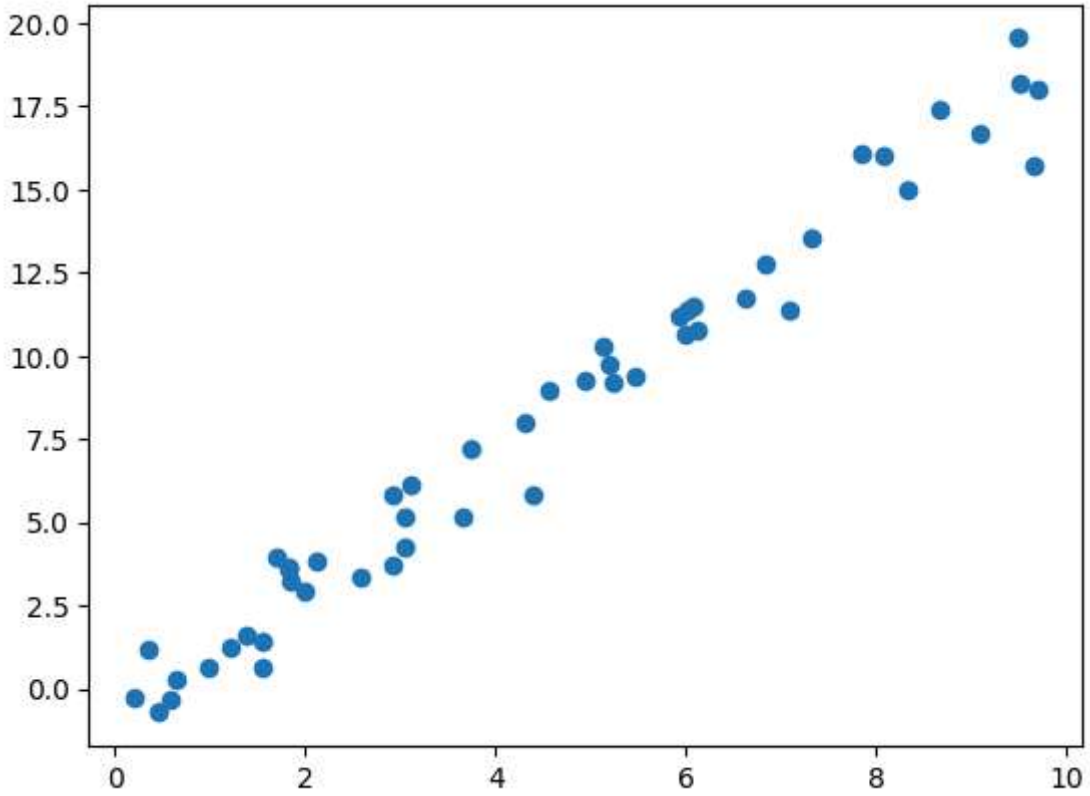
Простая линейная регрессия

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:
```
# при каждом запуске кода с seed (42) будут генерироватся одинаковые случайные числа
rng = np.random.RandomState(42)
x= 10 * rng.rand(50)
y = 2 * x - 1 + rng.randn(50)
plt.scatter(x,y);
```



In [16]:
```
# Шаг 1. Выбор линейной модели
from sklearn.linear_model import LinearRegression
```

In [17]:
```
# Шаг 2. Создание модели
model = LinearRegression(fit_intercept=True) # подбор точки пересечения с осью координат делаем
model
```

Out[17]:
```
LinearRegression()
```

In [ ]:
```
# Шаг 3. Формирование из данных матриц признаков и целевого вектора
x.shape
```

In [18]:
```
X = x[:,np.newaxis]
X.shape
```

Out[18]:
```
(50, 1)
```

In [19]:
```
# Шаг 4. Обучение модели на наших данных
model.fit(X,y)
```

Out[19]:
```
LinearRegression()
```

In [20]:
```
model.coef_ # угловой коэф.
```

Out[20]:
```
array([1.9776566])
```

In [21]:
```
model.intercept_ # точка пересечения с осью координат (верт)
```

Out[21]:
```
-0.903310725531111
```

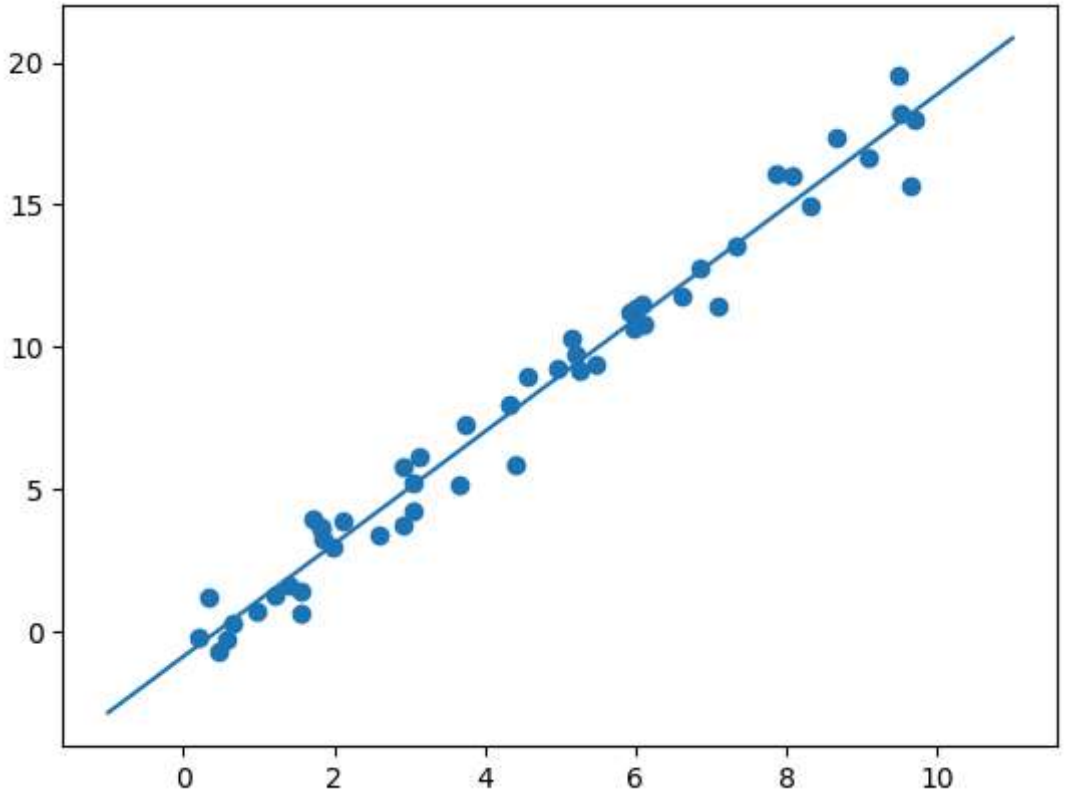
Эти два параметра представляют собой угловой коэффициент и точку пересечения с осью координат для простой линейной аппроксимации наших данных. Сравнивая с описанием данных, видим, что они очень близки к исходному угловому коэффициенту, равному 2, и точке пересечения, равной –1.

In [22]:
```
# Шаг 5. Предсказание меток для новых данных
xfit = np.linspace(-1,11)
xfit.shape
```

Out[22]:
```
(50,)
```

In [23]:
```
Xfit = xfit[:,np.newaxis]
yfit = model.predict(Xfit)
```

In [24]:
```
plt.scatter(x,y)
plt.plot(Xfit, yfit);
```



In [ ]: