Fourier Trigonometric Series

1. Write a Python function `FourierSeries(f,P,x,n)` that returns the values of the Fourier series of the function $f(x)$ at the point x. Plot the Fourier series of the following functions

In [72]:
```
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt


def plot_Fourier_function(f_latex, X, N, ax):
    legends = list()
    for i in range(len(X)):
        ax.set_title(f'Fourier series of ${f_latex}$', fontsize=20)
        ax.set_xlabel('$x$')
        ax.set_ylabel('$f(x)$')
        ax.plot(X[i])
        legends.append(f"$n={N[i]}$")
    ax.grid()
    ax.legend(legends, loc=2)


def obtain_a0(f, P):
    integral_ = integrate.quad(f, 0, P)
    return (1/P) * integral_[0]


def obtain_ak(f, P, k):
    f_ = lambda x: f(x) * np.cos((2*np.pi*k*x)/P)
    integral_ = integrate.quad(f_, 0, P)[0]
    return (2/P) * integral_


def obtain_bk(f, P, k):
    f_ = lambda x: f(x) * np.sin((2*np.pi*k*x)/P)
    integral_ = integrate.quad(f_, 0, P)[0]
    return (2/P) * integral_


def FourierSeries(f, P, x, n):
    a0 = obtain_a0(f, P)
    sum_ = sum([obtain_ak(f, P, k) * np.cos((2*k*np.pi*x)/P) +
                obtain_bk(f, P, k) * np.sin((2*k*np.pi*x)/P)
                for k in range(1, n+1)])

    return a0 + sum_
```

Fourier Series for $f := 1 - x$, $P = 1$ **and** $n = 6, 12, 36, 124$

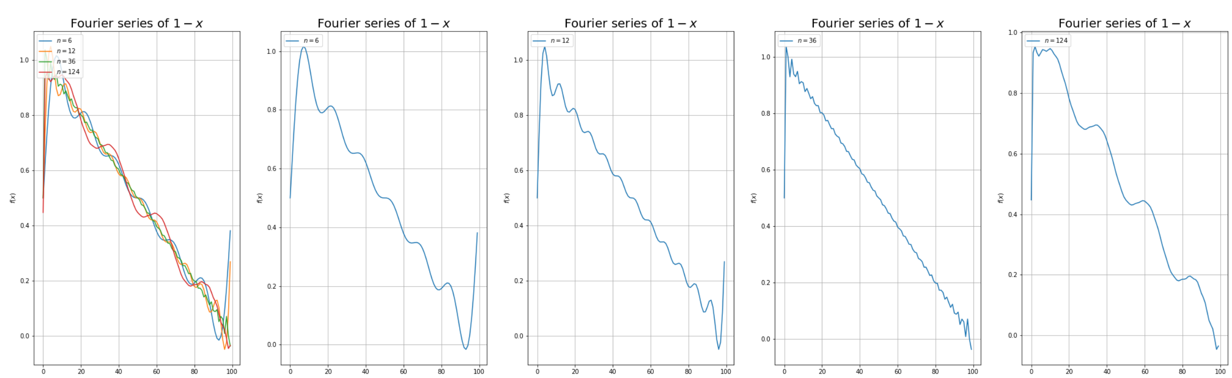
In [78]:
```python
P = 1
N_ = [6, 12, 36, 124]
X_ = np.arange(0, 1, 0.01)
series_list = list()

fig, axes = plt.subplots(nrows=1, ncols=5, figsize=(35,10))

f = lambda x: 1-x
for n in range(len(N_)):
    serie_sum = FourierSeries(f, P, X, N_[n])
    plot_Fourier_function('1 - x', [serie_sum], [N_[n]], axes[n+1])
    series_list.append(serie_sum)

# print(len(series_list))
plot_Fourier_function('1-x', series_list, N_, axes[0])
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:31: IntegrationWar
ning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:25: IntegrationWar
ning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:25: IntegrationWar
ning: The integral is probably divergent, or slowly convergent.
```



Fourier Series for $f := x^2, P = 1$ and $n = 6, 12, 36, 124$

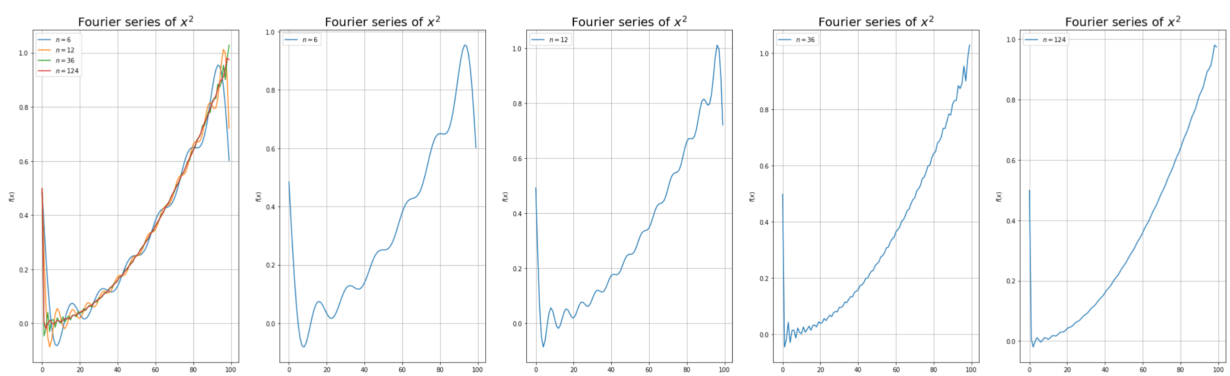
In [74]:
```python
P = 1
N_ = [6, 12, 36, 124]
X_ = np.arange(0, 1, 0.01)
series_list = list()

fig, axes = plt.subplots(nrows=1, ncols=5, figsize=(35,10))

f = lambda x: x**2
for n in range(len(N_)):
    serie_sum = FourierSeries(f, P, X, N_[n])
    plot_Fourier_function('x^2', [serie_sum], [N_[n]], axes[n+1])
    series_list.append(serie_sum)

# print(len(series_list))
plot_Fourier_function('x^2', series_list, N_, axes[0])
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:25: IntegrationWar
ning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:31: IntegrationWar
ning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
```



What do you observe as n becomes larger?

Answer:

As you can see in the above plot as n becomes larger the aproximation of the Fourier Series becomes better. However, the limitations of one CPU were also visible when $n = 124$. As you can see this lack of precision in the calculus made the plot of $f = x - 1$ when $n = 124$ worse than the one with $n = 36$.

2. Test the stability of the zero solution for the following system

$$x_1' = -4x_1 + 8x_1x_2^2$$

\

$$x_2' = -12x_1^2x_2 - 6x_2$$

Answer:

Consider $V(x_1, x_2) := ax_1^2 + bx_2^2$.

Then,

$$V' = 2ax_1(-4x_1 + 8x_1x_2^2) + 2bx_2(-12x_1^2x_2 - 6x_2)$$
$$V' = -8ax_1^2 + (16a - 24b)x_1^2x_2^2 - 12bx_2^2$$

If $a < 0$ and $b > 0$ then V' is negative and the system is stable.

If $16a - 24b = 0$ then V' is negative and the system is stable.

So, if one of the above conditions satifies the system would be stable.

In [3]:
```

```

In [3]:
```

```