Recursively split by character

This text splitter is the recommended one for generic text. It is parameterized by a list of characters. It tries to split on them in order until the chunks are small enough. The default list is `["\n\n", "\n", " ", ""]`. This has the effect of trying to keep all paragraphs (and then sentences, and then words) together as long as possible, as those would generically seem to be the strongest semantically related pieces of text.

1. How the text is split: by list of characters.
2. How the chunk size is measured: by number of characters.

```
# This is a long document we can split up.
with open("../../state_of_the_union.txt") as
f:
    state_of_the_union = f.read()
```

```
from langchain.text_splitter import
RecursiveCharacterTextSplitter
```

```
text_splitter =
RecursiveCharacterTextSplitter(
    # Set a really small chunk size, just to
show.
    chunk_size=100,
    chunk_overlap=20,
    length_function=len,
    is_separator_regex=False,
)
```

```
texts =
text_splitter.create_documents([state_of_the_un
print(texts[0])
print(texts[1])
```

```
page_content='Madam Speaker, Madam Vice
President, our First Lady and Second
Gentleman. Members of Congress and'
page_content='of Congress and the Cabinet.
Justices of the Supreme Court. My fellow
Americans.'
```

```
text_splitter.split_text(state_of_the_union)
[:2]
```

```
['Madam Speaker, Madam Vice President, our
First Lady and Second Gentleman. Members of
Congress and',
 'of Congress and the Cabinet. Justices of
the Supreme Court. My fellow Americans.']
```