

Argument

August 25, 2026

Argument

Kind: Class

Source: `lib/argument.js`

`Argument` defines a positional argument for a Commander command. It stores argument metadata such as the name, default value, parser, allowed choices, and whether the argument is required or optional.

Methods

Method	Signature	Returns
<code>name</code>	<code>name()</code>	<code>void</code>
<code>_collectValue</code>	<code>_collectValue(value: undefined, previous: undefined)</code>	<code>void</code>
<code>default</code>	<code>default(value: undefined, description: undefined)</code>	<code>void</code>
<code>argParser</code>	<code>argParser(fn: undefined)</code>	<code>void</code>
<code>choices</code>	<code>choices(values: undefined)</code>	<code>void</code>
<code>argRequired</code>	<code>argRequired()</code>	<code>void</code>
<code>argOptional</code>	<code>argOptional()</code>	<code>void</code>

Where it refuses work

- `Argument` stops the work with `InvalidArgumentError` when `!this.argChoices.includes(arg)` .
- `Argument` stops the work with an early return when `previous === this.defaultValue || !Array.isArray(previous)` .
- `Argument` stops the work with an early return when `this.variadic` .

Diagram

```
graph LR
  Command[Command] --> Argument[Argument]
  Argument --> Name[name()]
  Argument --> Default[default()]
  Argument --> Parser[argParser()]
  Argument --> Choices[choices()]
  Argument --> Required[argRequired()]
  Argument --> Optional[argOptional()]
  Parser --> Action[Command action]
  Choices --> Action
```

Usage

```
import { Argument, Command } from 'commander';

const mode = new Argument('mode')
  .choices(['development', 'production'])
  .argRequired();

const label = new Argument('label')
  .default('local')
  .argOptional();

const program = new Command();

program
  .name('deploy')
  .addArgument(mode)
  .addArgument(label)
  .action((selectedMode, selectedLabel) => {
    console.log(`Deploying ${selectedLabel} in ${selectedMode} mode`);
  });

program.parse();
```

API Coding Instructions

- Create `Argument` instances and register them with a `Command` using `addArgument()`.
- Use `argRequired()` or `argOptional()` to set positional argument behavior when it is not defined by the argument syntax.
- Use `choices()` for fixed string values; it sets the argument parsing behavior for choice validation.
- Use `argParser()` when an argument needs value conversion or custom validation, and return the parsed value.

- Treat `_collectValue()` as internal behavior for collecting repeated values rather than calling it from command code.

Relationships

- IMPORTS → `InvalidArgumentError`

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `program` — `index.js :7`
- `Command` — `lib/command.js :14`