

Задайте цвета графика

MATLAB® создает графики с помощью набора по умолчанию цветов. Цвета по умолчанию обеспечивают чистый и сопоставимый взгляд через различные графики, которые вы создаете. Можно настроить цвета, если вы должны. Много функций построения графика имеют входной параметр, такой как `c` или `colspec` для настройки цвета. Объекты, возвращенные этими функциями обычно, имеют свойства для управления цветом. Имена аргументов и свойств могут варьироваться, но значения, которые они принимают обычно, следуют за общим шаблоном. Если вы знакомы с шаблоном, можно использовать его, чтобы изменить большое разнообразие графиков.

Типы значений цвета

Существуют эти типы значений цвета:






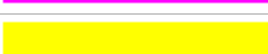


- **Название цвета или Краткое название** — Задают имя цвета, такого как `'red'` или `'green'`. Краткие названия задают букву с названия цвета, такого как `'r'` или `'g'`.
- **Триплет RGB** — Создает пользовательский цвет путем определения трехэлементного вектора-строки, элементами которого является интенсивность красных, зеленых, и синих компонентов цвета. Интенсивность должна быть в области значений $[0,1]$. Например, можно задать оттенок розового как `[1 0.5 0.8]`.

Аргументы функции, которые управляют цветом, не всегда поддерживают триплеты RGB, но свойства объектов, которые управляют цветом обычно, делают.








- **Шестнадцатеричный Цветовой код (*Начиная с R2019a*)** — Создает пользовательский цвет путем определения вектора символов или строкового скаляра, который запускается с символа хеша (`#`) сопровождаемый тремя или шестью шестнадцатеричными цифрами, которые могут лежать в диапазоне от 0 к F. Значения не являются чувствительными к регистру. Таким образом, цветовые коды `'#FF8800'`, `'#ff8800'`, `'#F80'`, и `'#f80'` все задают тот же оттенок оранжевого.

Аргументы функции, которые управляют цветом, обычно не поддерживают шестнадцатеричные цветовые коды, но свойства объектов, которые управляют цветом обычно, делают.

Эта таблица приводит все допустимые названия цвета и краткие названия с соответствующими триплетами RGB и шестнадцатеричными цветовыми кодами.

Название цвета	Краткое название	Триплет RGB	Шестнадцатеричный цветовой код	Внешний вид
'red'	'r'	[1 0 0]	'#FF0000'	
'green'	'g'	[0 1 0]	'#00FF00'	
'blue'	'b'	[0 0 1]	'#0000FF'	
'cyan'	'c'	[0 1 1]	'#00FFFF'	
'magenta'	'm'	[1 0 1]	'#FF00FF'	
'yellow'	'y'	[1 1 0]	'#FFFF00'	
'black'	'k'	[0 0 0]	'#000000'	
'white'	'w'	[1 1 1]	'#FFFFFF'	

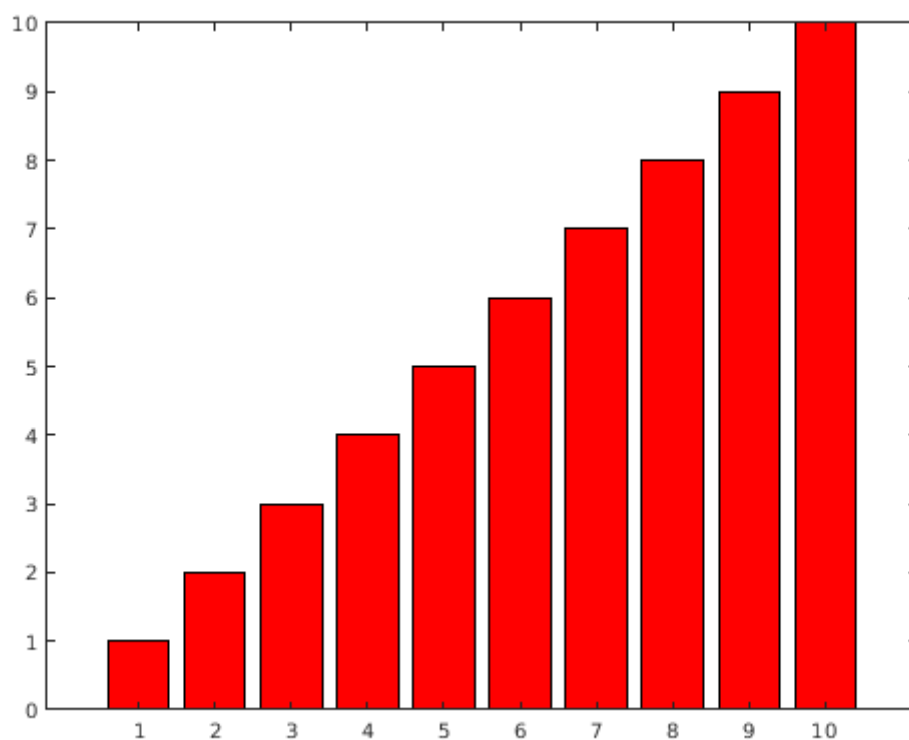
Вот являются триплеты RGB и шестнадцатеричные цветовые коды для цветов по умолчанию использованием MATLAB во многих типах графиков. Эти цвета не имеют имен сопоставленными с ними.

Триплет RGB	Шестнадцатеричный цветовой код	Внешний вид
[0.4470 0.7410]	'#0072BD'	
[0.8500 0.3250 0.0980]	'#D95319'	
[0.9290 0.6940 0.1250]	'#EDB120'	
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

Задайте цвет столбчатой диаграммы

Создайте красную столбчатую диаграмму путем вызова `bar` функция и определение дополнительного `color` аргумент как `'red'`. Возвратите объект панели как `b`, таким образом, можно настроить другие аспекты графика позже.

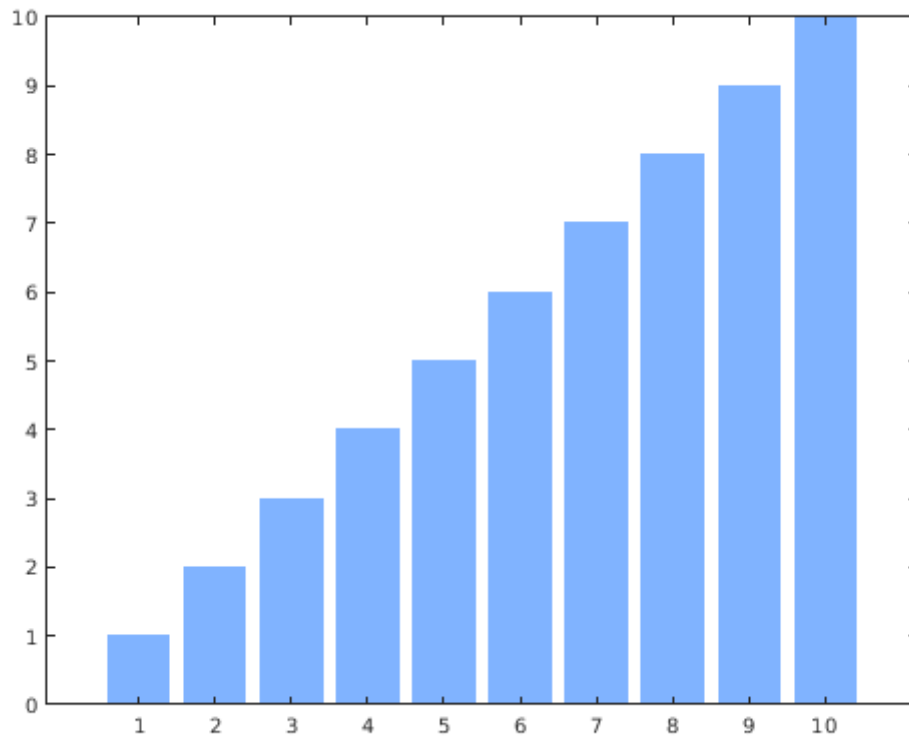
```
b = bar(1:10, 'red');
```



Теперь измените цвет заливки панели и цвет контура к голубому путем установки `FaceColor` и `EdgeColor` свойства к шестнадцатеричному цветовому коду, `'#80B3FF'`.

Перед R2019a задайте триплет *RGB* вместо шестнадцатеричного цветового кода. Например, `b.FaceColor = [0.5 0.7 1]`.

```
b.FaceColor = '#80B3FF';  
b.EdgeColor = '#80B3FF';
```



Задайте цвета маркера в графике поля точек

Создайте график рассеивания случайных чисел. Задайте размер маркера как 75 точки и аргументы `name-value` использования, чтобы задать цвета контура маркера и цвета заливки. `MarkerEdgeColor` свойство управляет цветом контура и `MarkerFaceColor` управляет цветом заливки.

```
x = rand(1,100);  
y = rand(1,100);  
scatter(x,y,75,'MarkerEdgeColor','b', ...  
        'MarkerFaceColor',[0 0.7 0.7])
```

