

ad-6. スタック

(C 言語によるアルゴリズムとデータ構造) (全 6 回)

URL: <https://www.kkaneko.jp/pro/ad/index.html>

金子邦彦



アウトライン



6-1 スタック, プッシュとポップ

- ① **ウェブブラウザ**を起動する
- ② 次の URL を開く

<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>

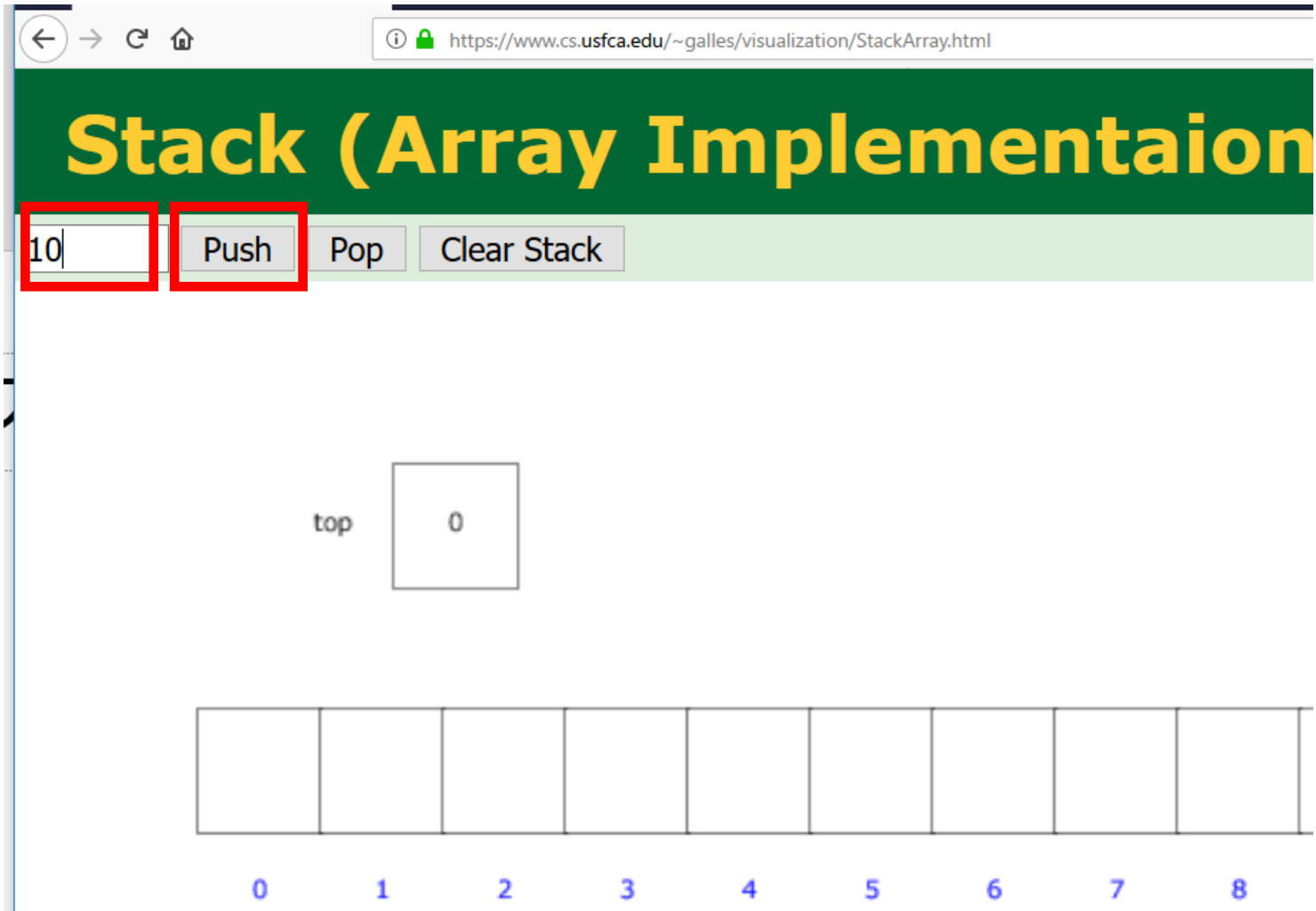
米国サンフランシスコ大の中のサイト

③ 「Stack: Array Implementation」 をクリック

Currently, we have visualizations for the following

- Basics
 - **Stack: Array Implementation**
 - Stack: Linked List Implementation
 - Queues: Array Implementation
 - Queues: Linked List Implementation
 - Lists: Array Implementation (available)
 - Lists: Linked List Implementation (available)
- Recursion
 - Factorial
 - Reversing a String
 - N-Queens Problem

④ 試しに「**10**」を入れ、「**Push**」クリック



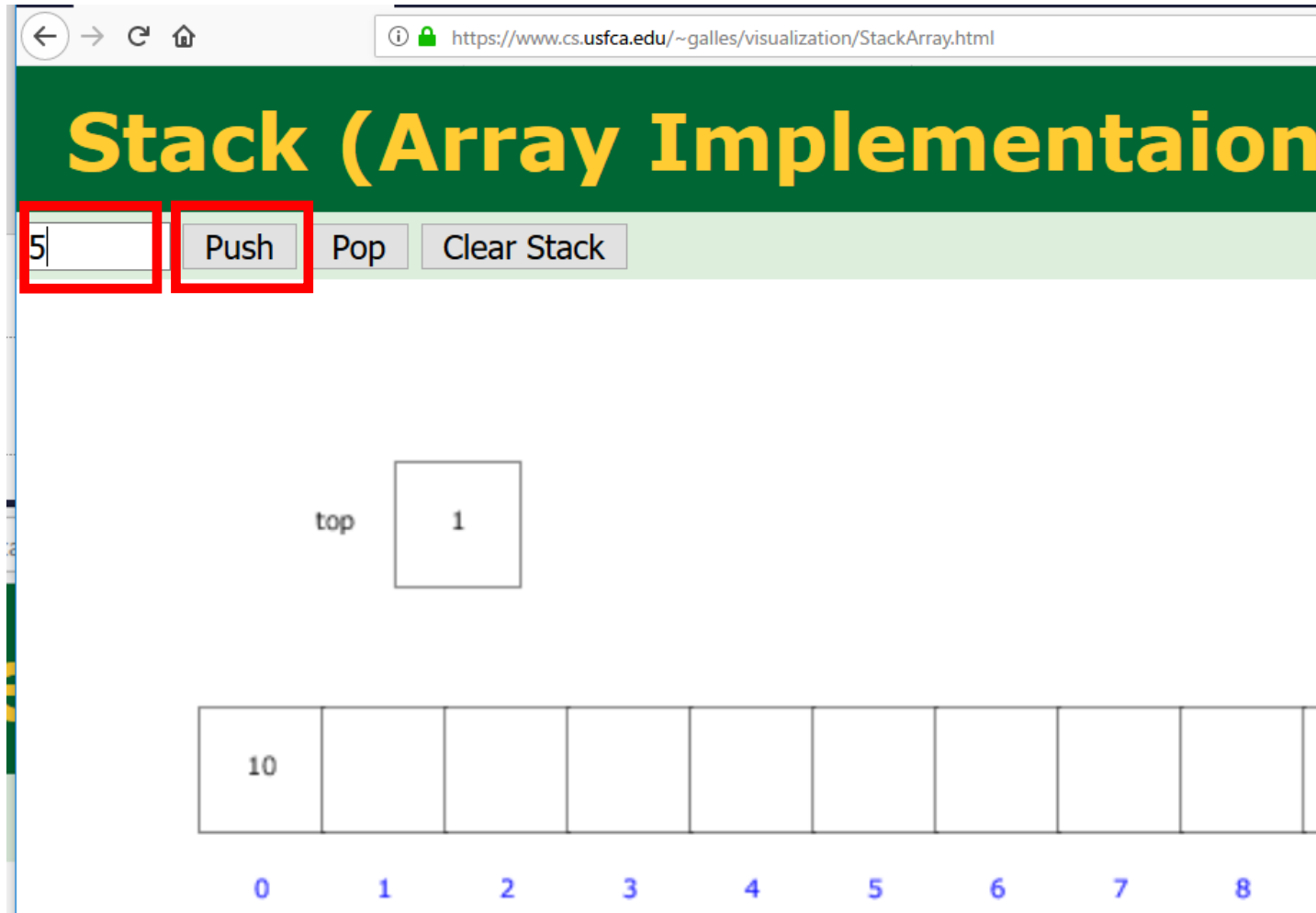
Stack (Array Implementation)

10 | Push | Pop | Clear Stack

top 0

0 1 2 3 4 5 6 7 8

⑤ 次に「5」を入れ、「Push」をクリック



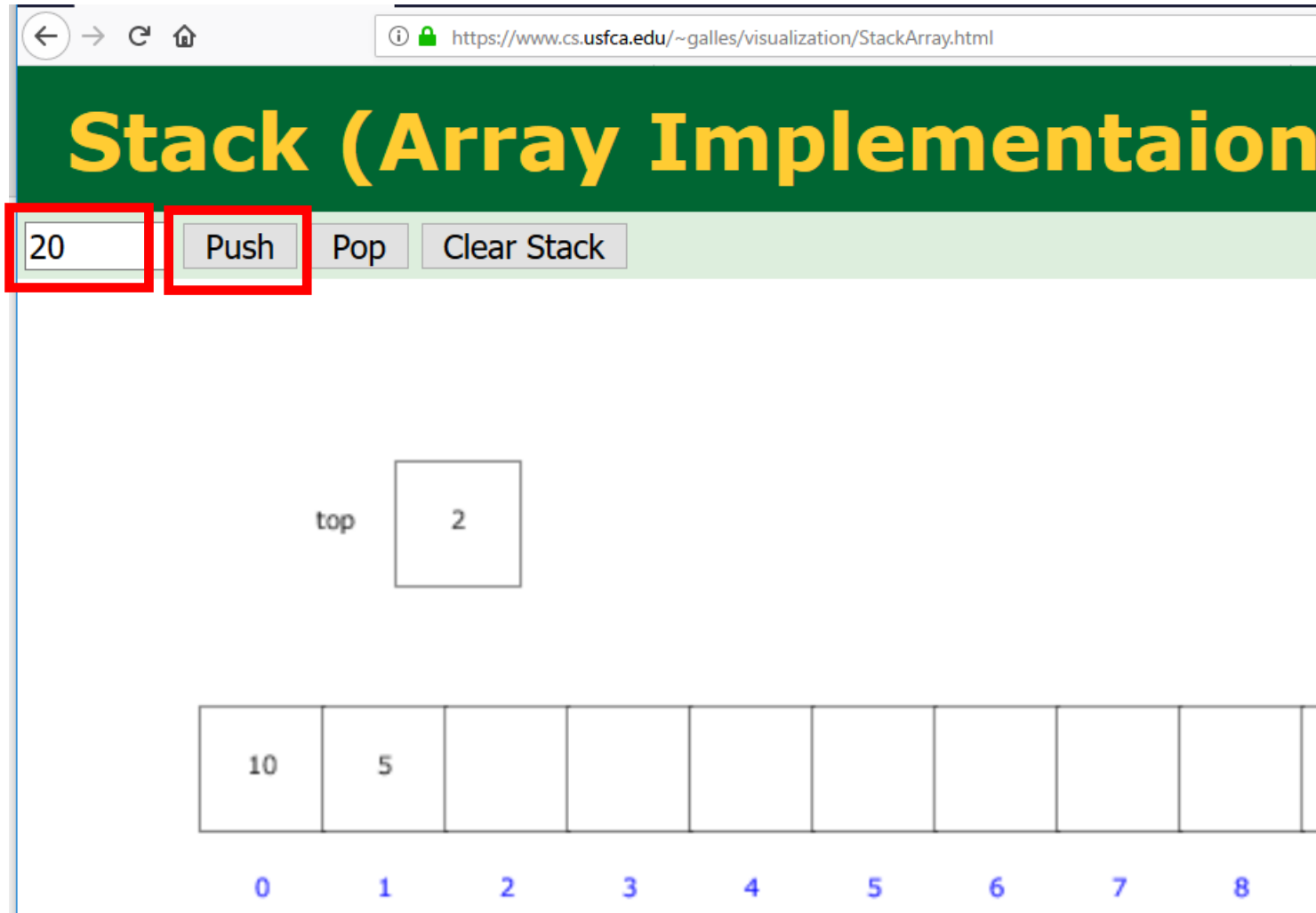
Stack (Array Implementaion)

5 | Push | Pop | Clear Stack

top 1

10									
0	1	2	3	4	5	6	7	8	

⑥ 次に「20」を入れ、「Push」をクリック

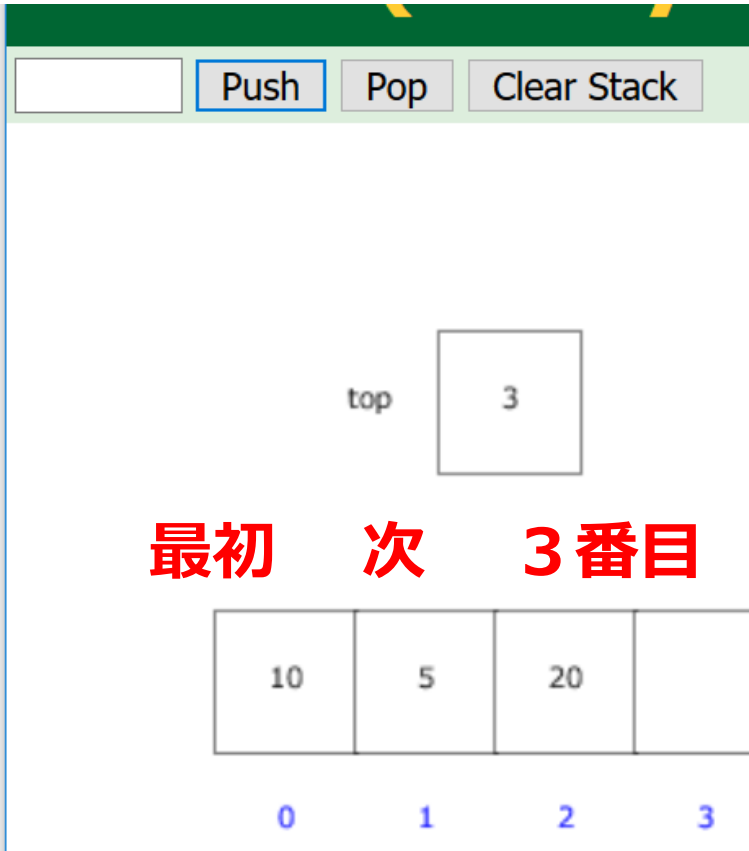


Stack (Array Implementaion)

20 Push Pop Clear Stack

top 2

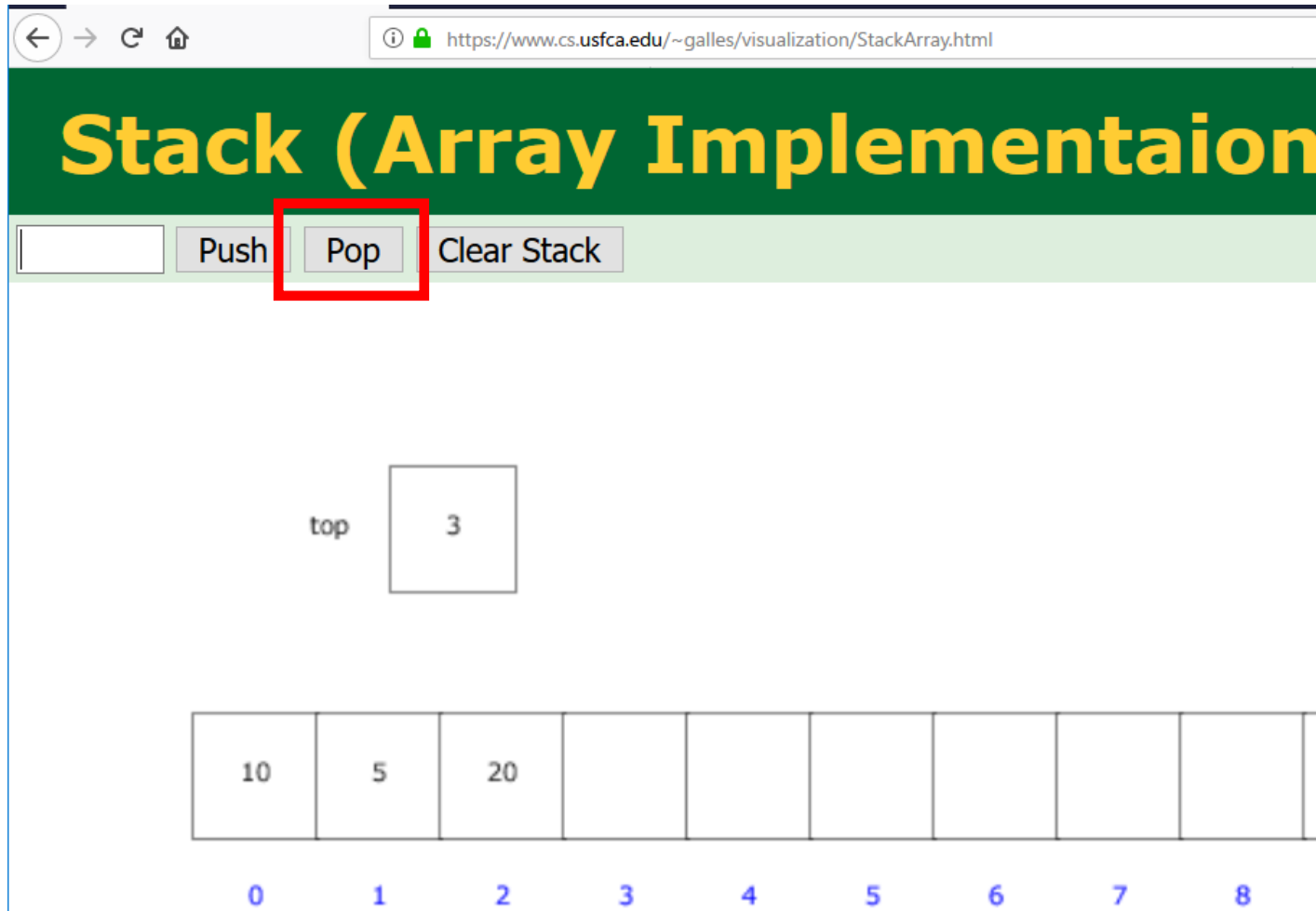
10	5							
0	1	2	3	4	5	6	7	8



**3つのデータ
10, 5, 20 が格納された**

※ **スタック**は
複数のデータを格納できる
データ構造

⑦ 今度は「**Pop**」をクリック。「**20**」が出てくる



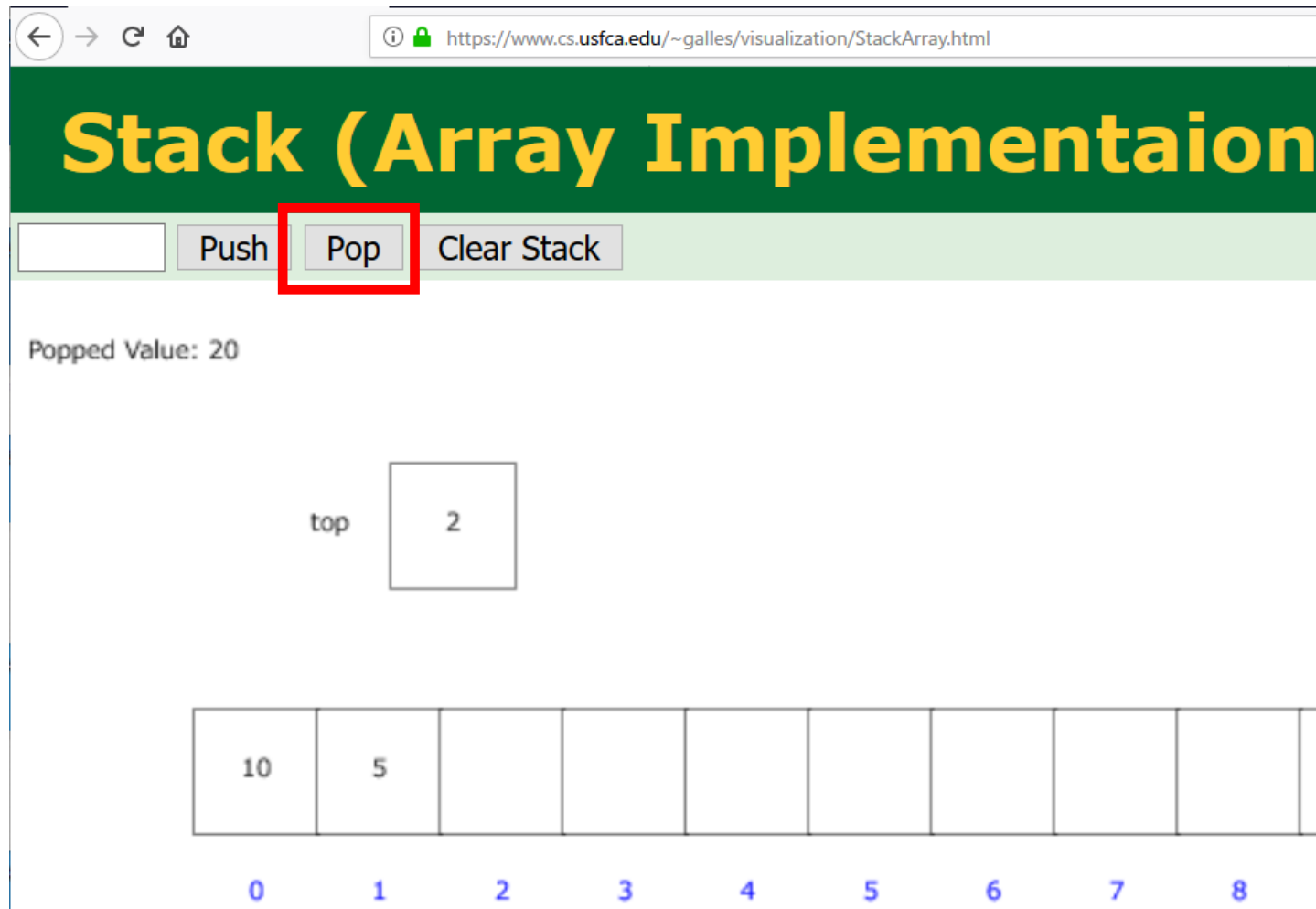
Stack (Array Implementaion)

Push Pop Clear Stack

top 3

10	5	20						
0	1	2	3	4	5	6	7	8

⑧ 「Pop」 をクリック. 「5」 が出てくる



Stack (Array Implementaion)

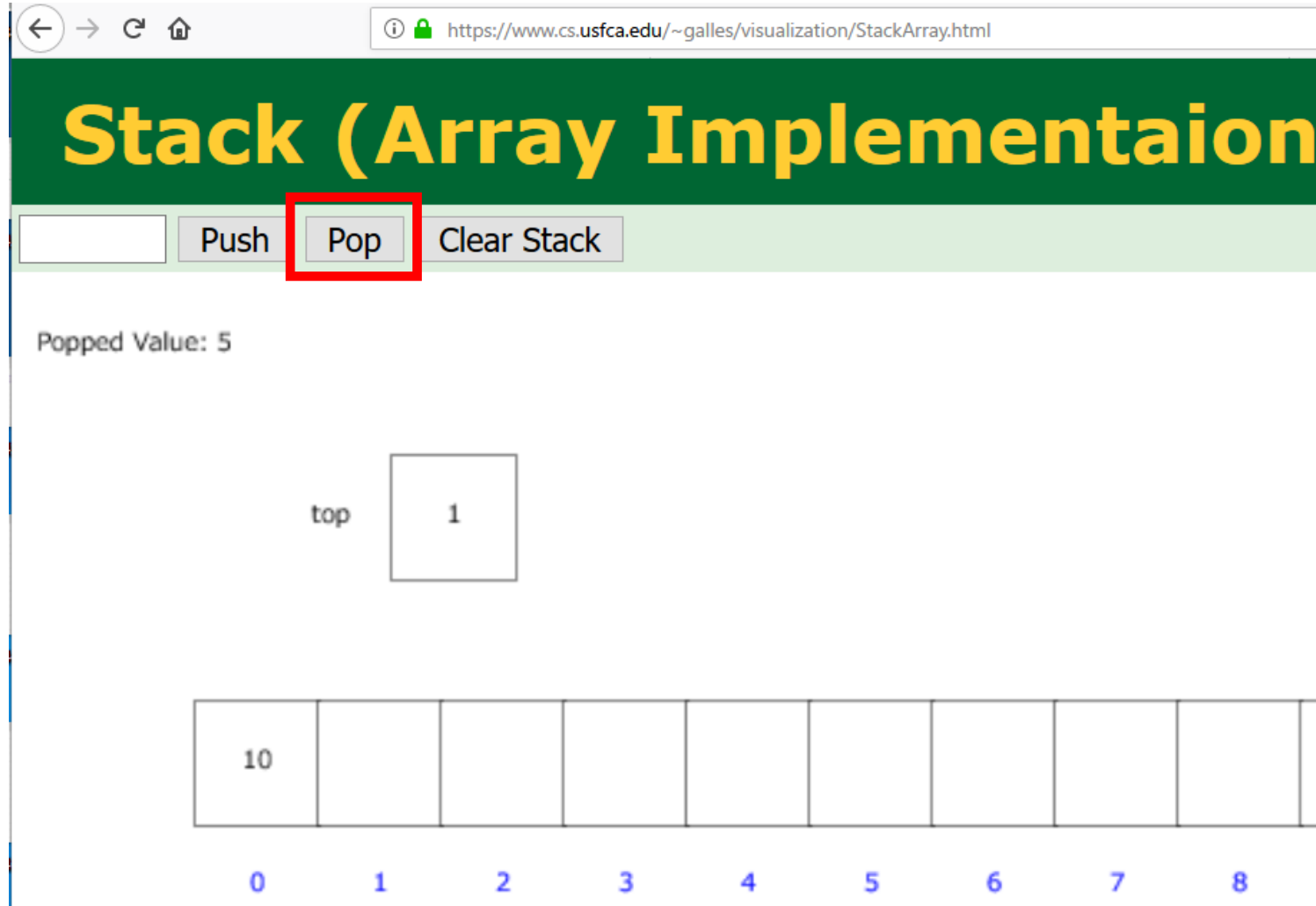
Push Pop Clear Stack

Popped Value: 20

top 2

10	5								
0	1	2	3	4	5	6	7	8	

⑨ 「Pop」 をクリック. 「10」 が出てくる



Stack (Array Implementaion)

Push Pop Clear Stack

Popped Value: 5

top 1

10								
0	1	2	3	4	5	6	7	8

スタックのプッシュとポップ



次にプッシュできる
場所を示している



- **プッシュ** :
スタックの一番上に追加
- **ポップ** :
スタックの一番上から削除

スタックの構成



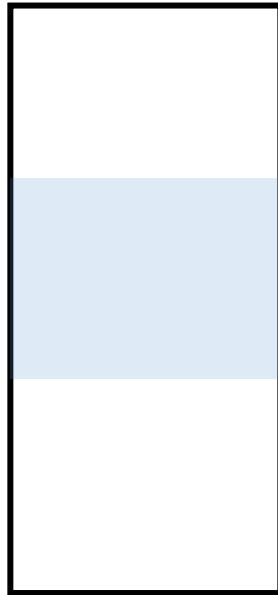
配列を使用する場合



事前に、メモリ内に
配列を確保

メモリ

スタックの構成



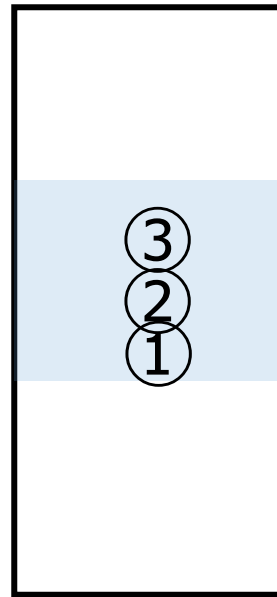
最初は空

メモリ

スタックの構成



①、②、③の順で
プッシュすると



メモリ

一番最後にプッシュ
した③が一番上

データが入っていく