

# CSc 110, Autumn 2017

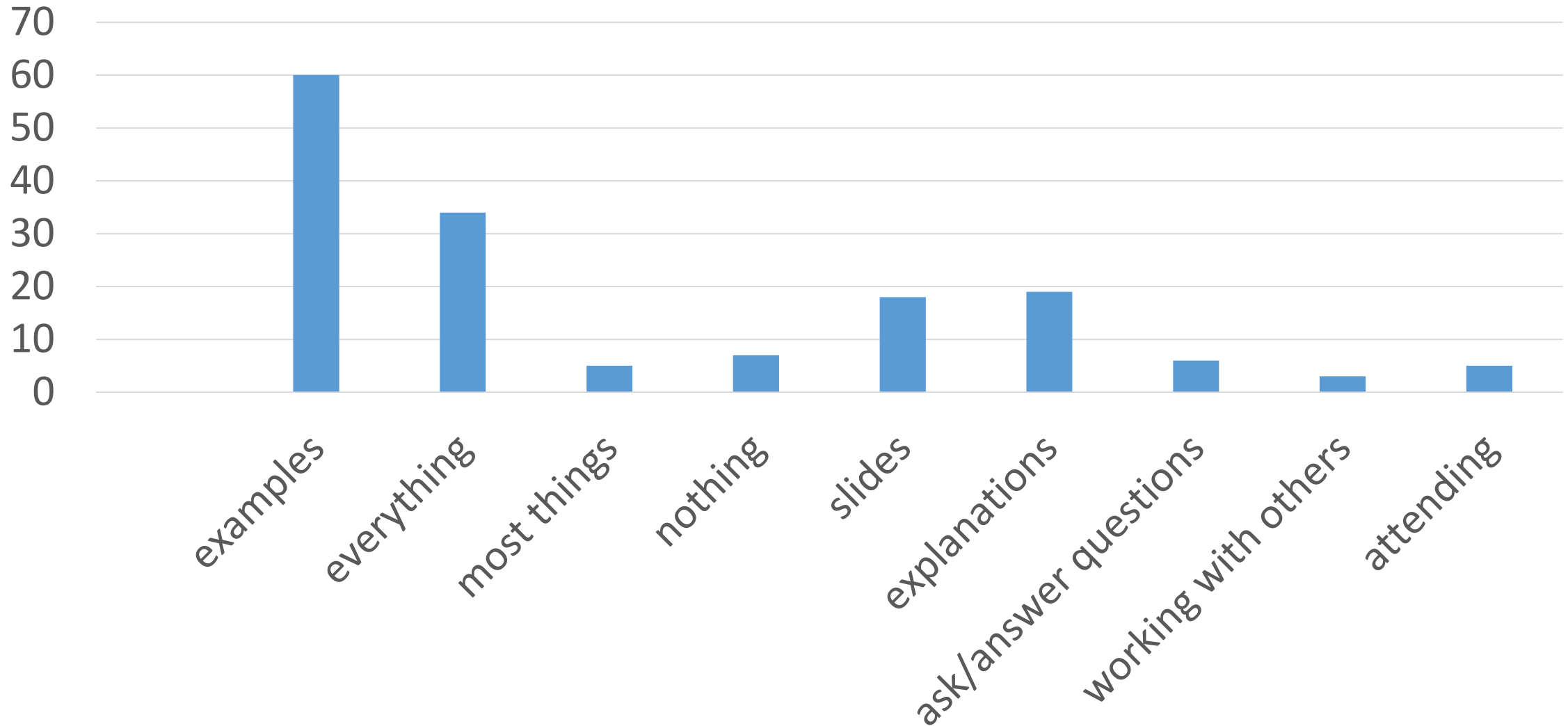
## Lecture 22: lists

Adapted from slides by Marty Stepp and Stuart Reges

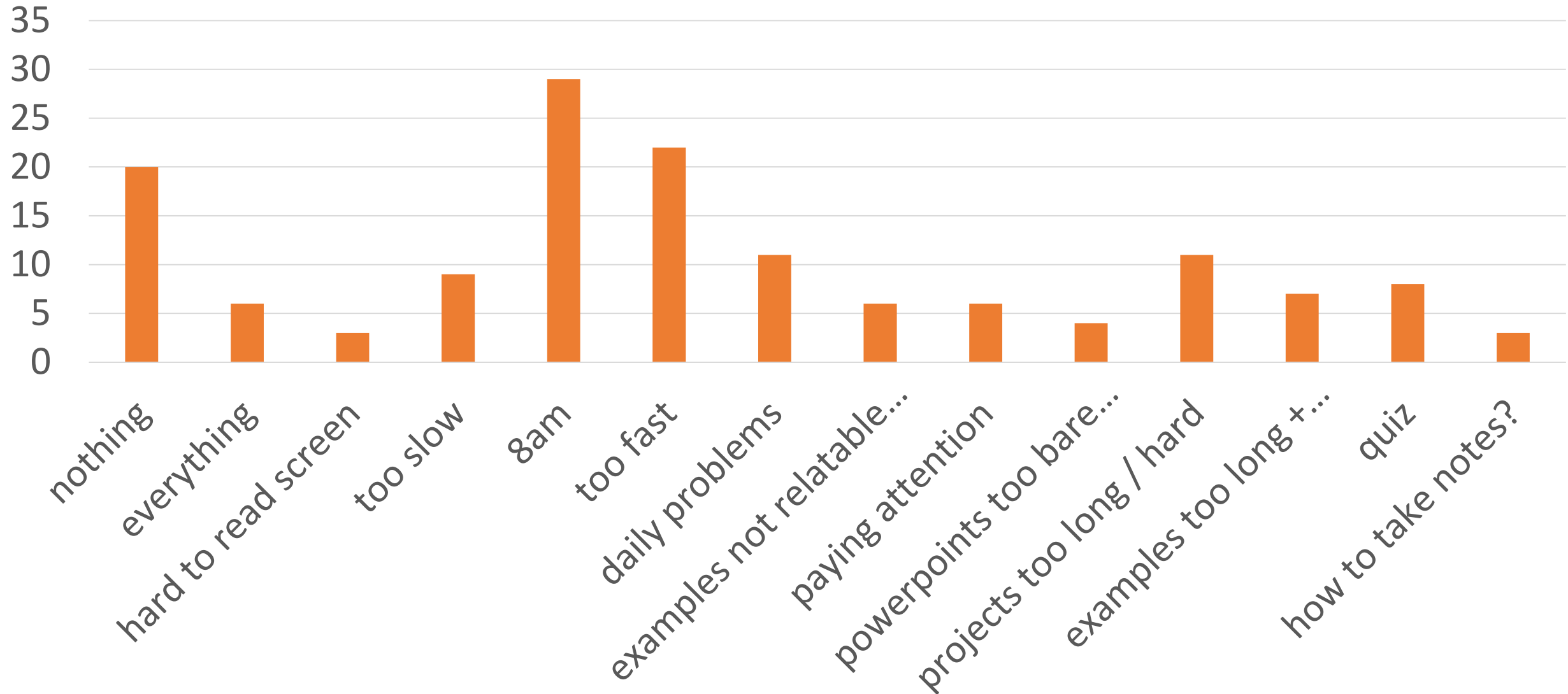


“The machine learning algorithm wants to know if we’d like a dozen wireless mice to feed the Python book we just bought.”

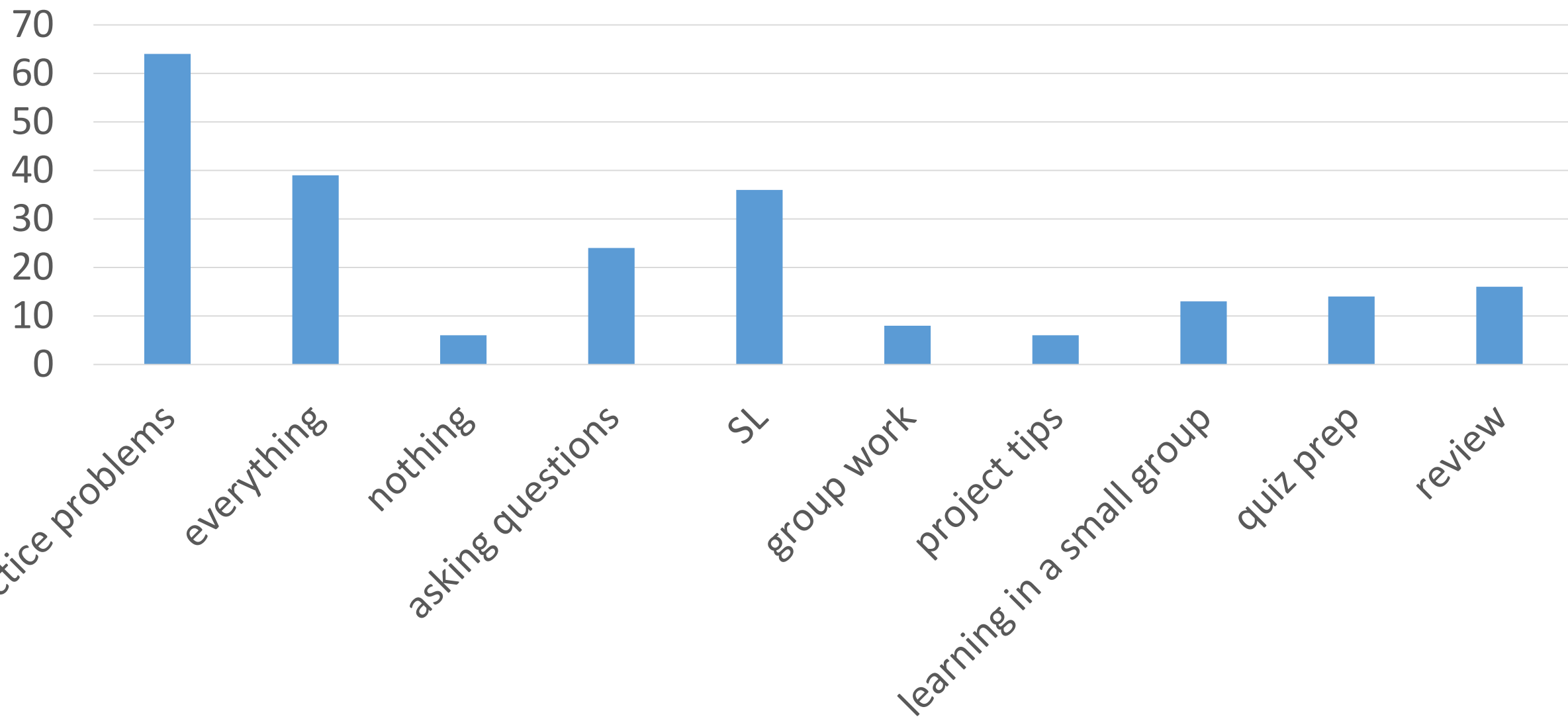
# What you like about lecture



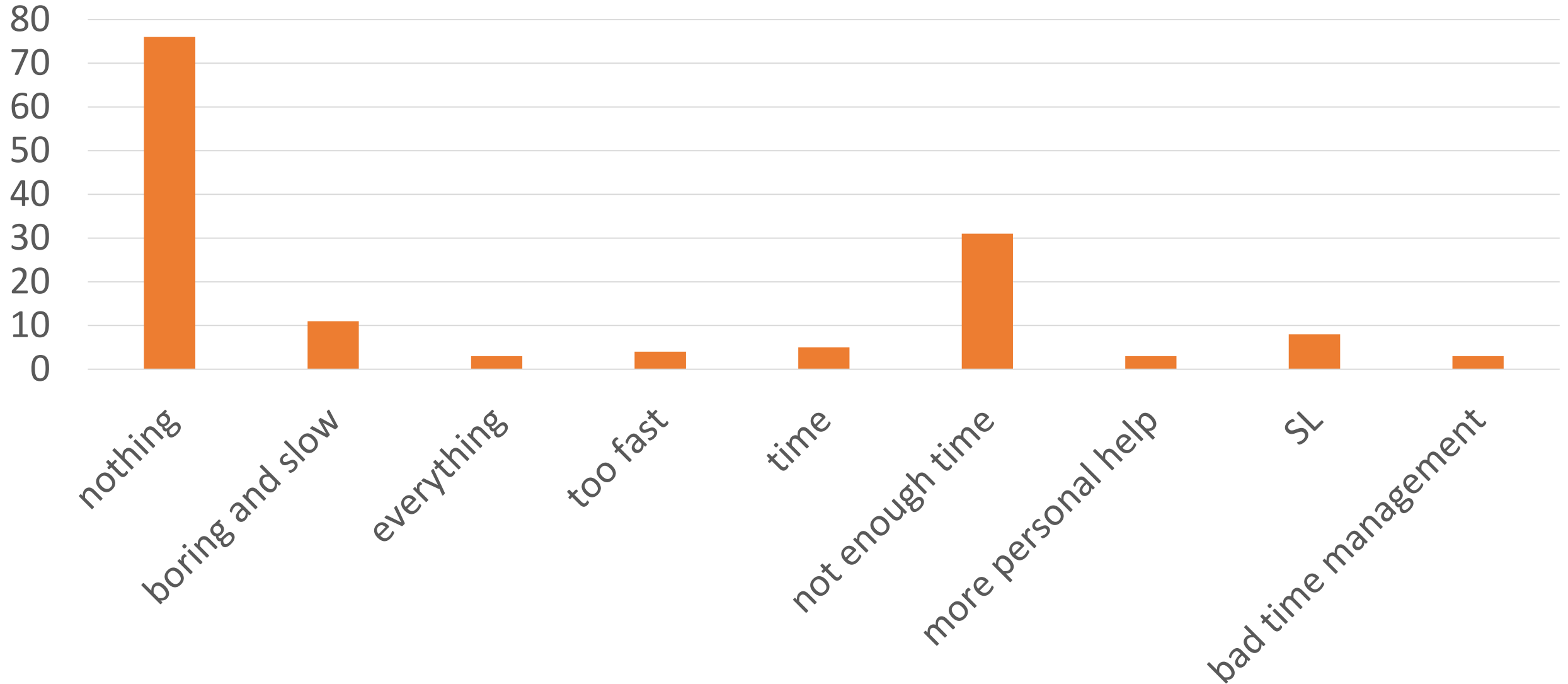
# What you dislike about lecture



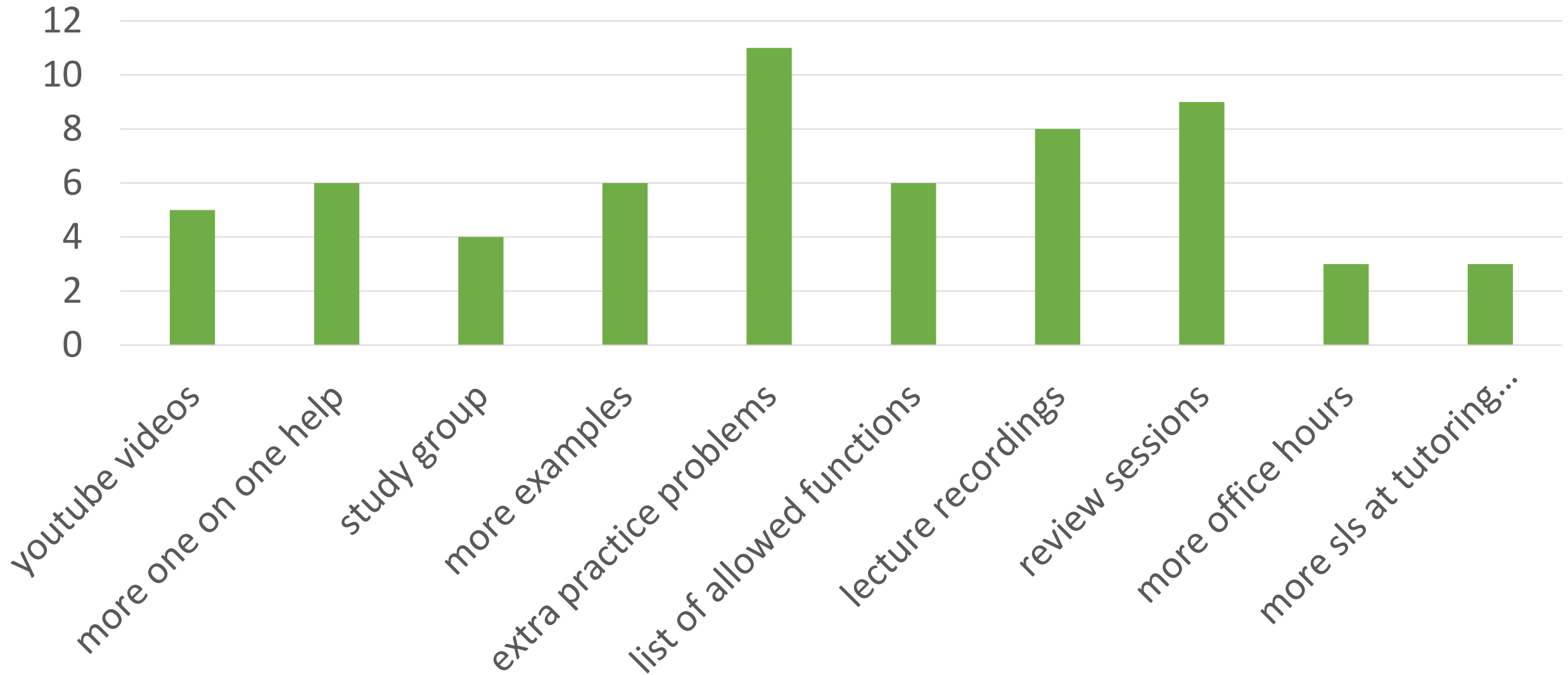
# What you like about section



# What you dislike about section



# Resources you would like



## Least favorite

- Handwritten exams
- Quizzes
- Time of deadlines
- Daily problems
- No collaboration

## Favorite

- Problem solving
- Joy of getting a problem right
- Programming
- Resources
- Projects
- Cartoons

"The projects are probably one of the most satisfying pieces of schoolwork I've ever done. I just feel a real sense of accomplishment when I complete one."

# Lists

- **list**: object that stores many values.
  - **element**: One value in a list.
  - **index**: A 0-based integer to access an element from an list.

<i>index</i>	0	1	2	3	4	5	6	7	8	9
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
<i>value</i>	12	49	-2	26	5	17	-6	84	72	3

↑	↑	↑
element 0	element 4	element 9



# List initialization

**name** = [**value**, **value**, ... **value**]

- Example:

```
numbers = [12, 49, -2, 26, 5, 17, -6]
```

*index* 0 1 2 3 4 5 6

*value*

12	49	-2	26	5	17	-6
----	----	----	----	---	----	----

- Useful when you know what the list's elements will be

**name** = [**value**] \* **count**

- Example:

```
numbers = [0] * 4
```

*index* 0 1 2 3

*value*

0	0	0	0
---	---	---	---

# List initialization

**name** = [**value**] \* **size**

- Example:

```
numbers = [0] * 3
```

Creates the following list

*index*

<i>value</i>	2	4	26
--------------	---	---	----

# Accessing elements

**name [index]**                   # **access**

**name [index] = value**       # **modify**

- Example:

```
numbers = [0] * 2
```

```
numbers[0] = 27
```

```
numbers[1] = -6
```

```
print(numbers[0])
```

```
if (numbers[1] < 0):
```

```
    print("Element 1 is negative.")
```

*index*   0   1

*value*   **27** | **-6**



# Lists and `for` loops

- It is common to use `for` loops to access list elements.

```
for i in range(0, 8):  
    print(str(numbers[i]) + " ", end='')  
print() # output: 0 4 11 0 44 0 0 2
```

- Sometimes we assign each element a value in a loop.

```
for i in range(0, 8):  
    numbers[i] = 2 * i
```

*index* 0 1 2 3 4 5 6 7

<i>value</i>	0	2	4	6	8	10	12	14
--------------	---	---	---	---	---	----	----	----

# len()

- Use `len()` to find the number of elements in a list.

```
for i in range(0, len(numbers)):  
    print(numbers[i] + " ", end='')  
# output: 0 2 4 6 8 10 12 14
```

- What expressions refer to:
  - The last element of any list?
  - The middle element?

# Lists and `for` loops

- You can also loop directly over lists, just as with strings

```
list = [1, 3, 6, 23, 43, 12]
for number in list:
    print(str(number + " "), end='')
print() # output: 1 3 6 23 43 12
```

# List functions

Function	Description
<code>append(x)</code>	Add an item to the end of the list. Equivalent to <code>a[len(a):] = [x]</code> .
<code>extend(L)</code>	Extend the list by appending all the items in the given list. Equivalent to <code>a[len(a):] = L</code>
<code>insert(i, x)</code>	Inserts an item at a given position. <code>i</code> is the index of the element before which to insert, so <code>a.insert(0, x)</code> inserts at the front of the list.
<code>remove(x)</code>	Removes the first item from the list whose value is <code>x</code> . Errs if there is no such item.
<code>pop(i)</code>	Removes the item at the given position in the list, and returns it. <code>a.pop()</code> removes and returns the last item in the list.
<code>clear()</code>	Remove all items from the list.
<code>index(x)</code>	Returns the index in the list of the first item whose value is <code>x</code> . Errs if there is no such item.
<code>count(x)</code>	Returns the number of times <code>x</code> appears in the list.
<code>sort()</code>	Sort the items of the list
<code>reverse()</code>	Reverses the elements of the list
<code>copy()</code>	Return a copy of the list.



# "list mystery" problem

- **traversal:** An examination of each element of an list.
- What element values are stored in the following list?

```
a = [1, 7, 5, 6, 4, 14, 11]
for i in range(0, len(a) - 1):
    if (a[i] > a[i + 1]):
        a[i + 1] = a[i + 1] * 2
```

<i>index</i>	0	1	2	3	4	5	6
<i>value</i>	1	7	10	12	8	14	22