

# 情報メディア 基盤ユニット

---

文字列とファイルの読み書き

情報メディア学科佐藤尚

# 最終課題制作のお知らせ

---

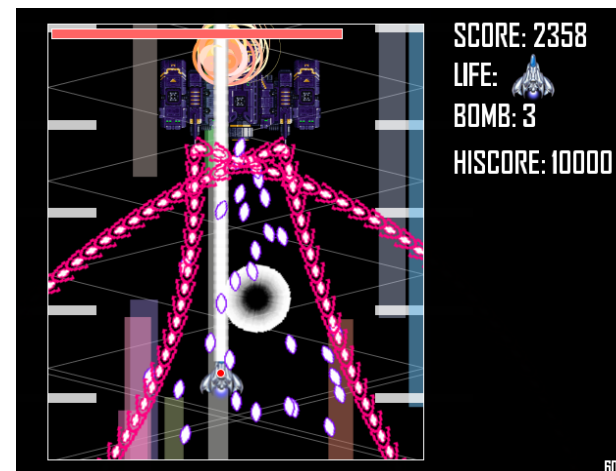
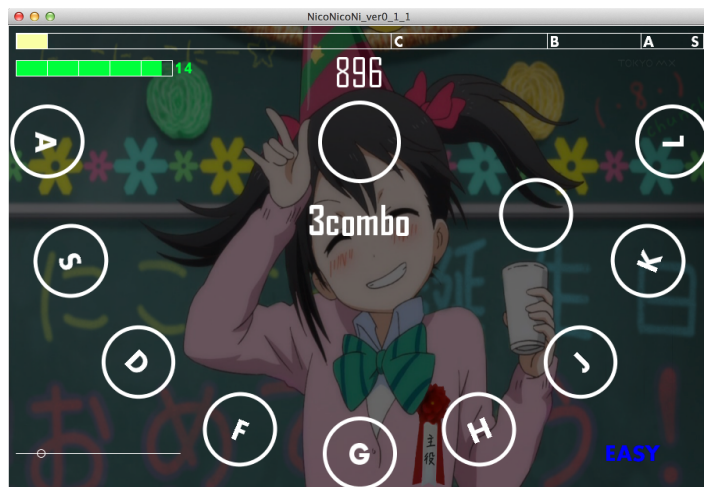
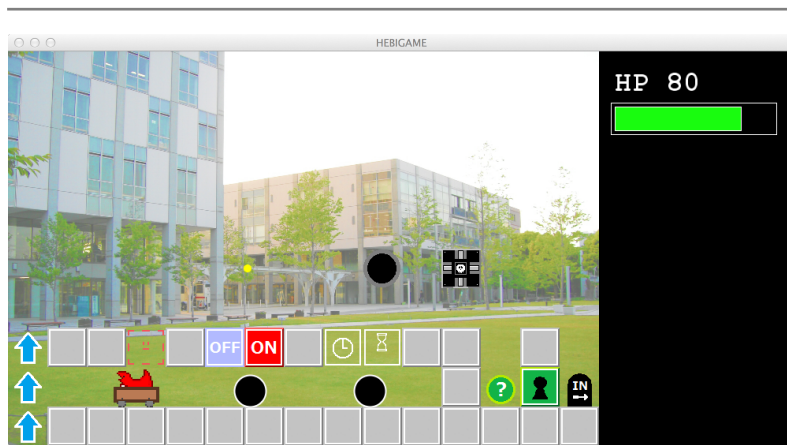
## 提出物

- Processingで作ったプログラム
  - （「スケッチをアーカイブ」の機能を利用して、zipファイルとして提出）
- 自分の制作物の内容を説明するWordで作ったレポート
- 5分間のプレゼンテーションを想定した、自分の作品紹介用のPowerPoint

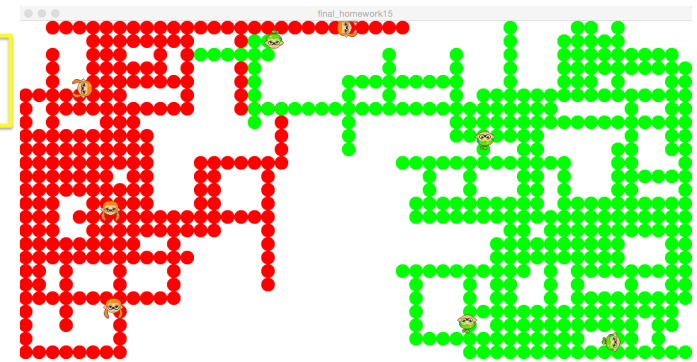
提出締め切り：7月26日（火）の2限授業開始時まで

発表会：7月29日（金）の演習時

# 2014年度1年前期の最終課題



# 2015年度1年生前期の最終課題



# String と char

---

String

色々な長さの文字列を扱う  
“abc”, “Kait 2016”, “!”, “”

参照型

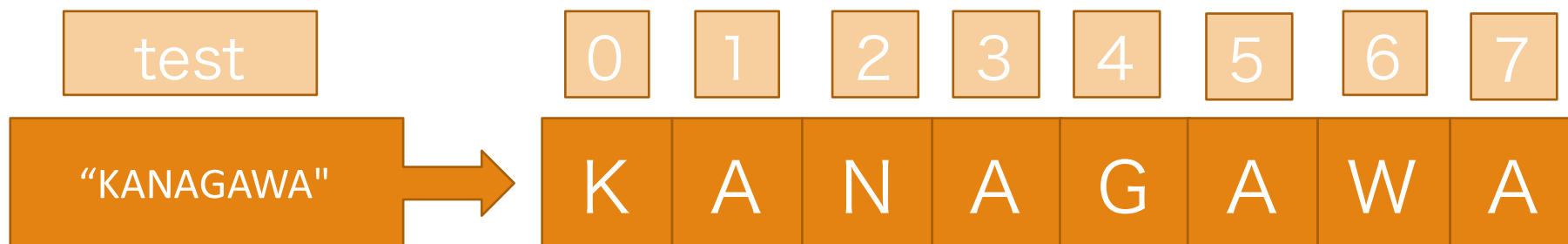
char

一文字だけの情報を表す  
'a', 'B', '!', '1'

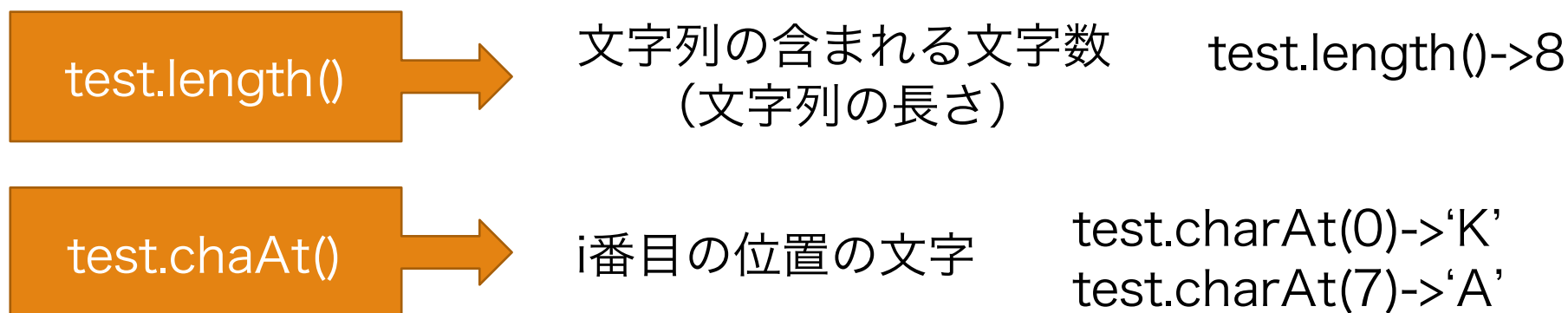
character

# 文字列(String)

---

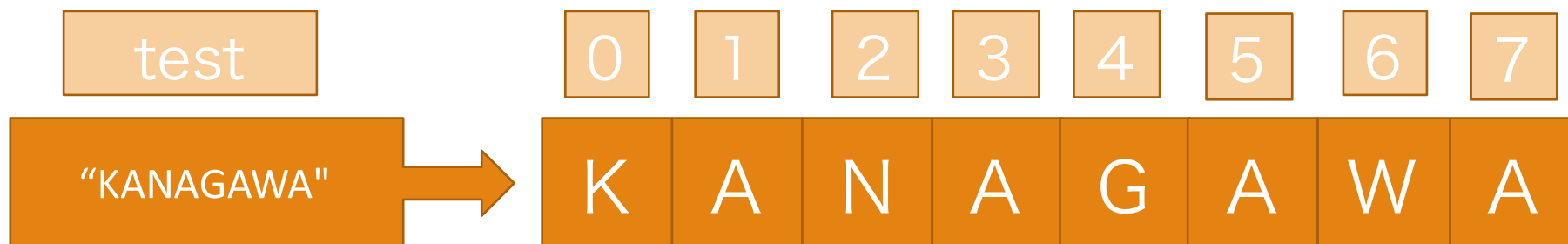


```
String test="KANAGAWA";
```

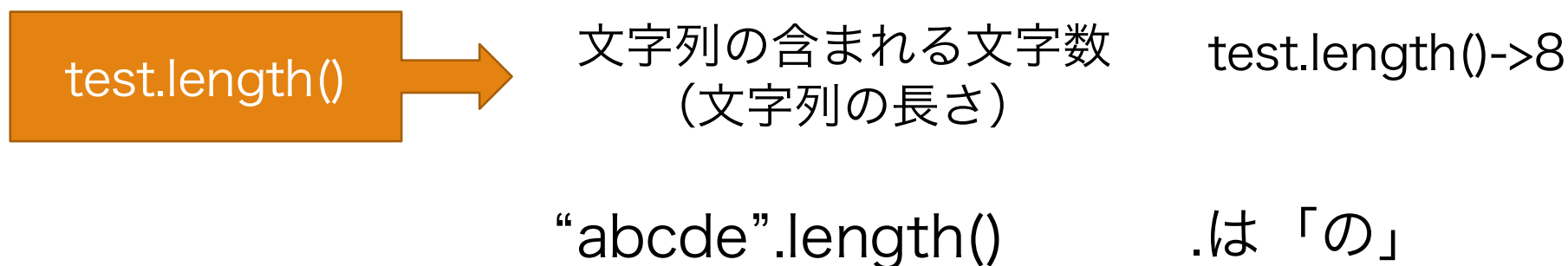


# 文字列(String)

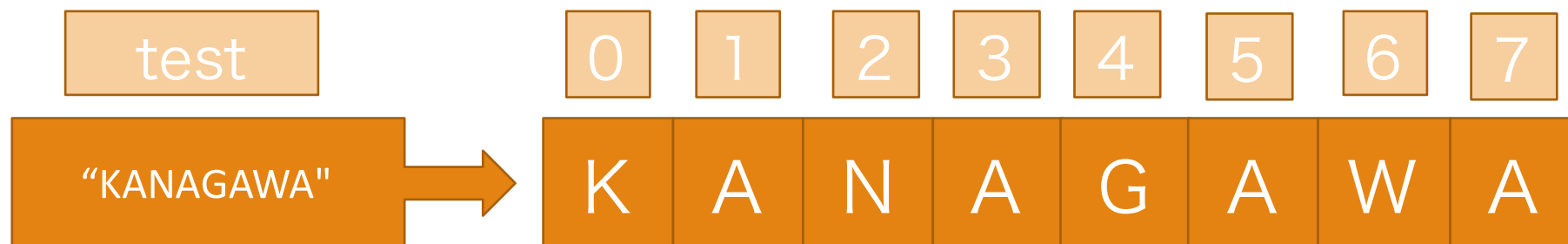
---



```
String test="KANAGAWA";
```



# 文字列(String)



```
String test="KANAGAWA";
```

```
test.indexOf("A")->1
```

```
test.indexOf("N")->2
```

```
test.indexOf("NA")->2
```

```
test.indexOf("IT")->-1
```

```
test.indexOf()
```

引数で指定した文字列が  
最初に現れる位置

indexOf(探す文字列、探し始める位置)

```
test.indexOf("A",2) ->3
```

指定した文字列が見つからなければ、戻り値は-1となる



# 例えば、

---

String alphabet = “abcdefghijklmnopqrstuvwxyz”;

alphabet.charAt(i)



i番目のアルファベットを取り出す

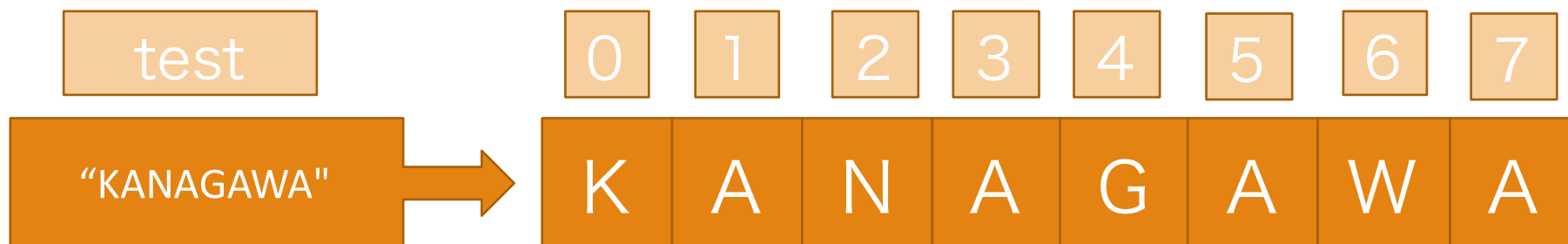
alphabet.indexOf(alpha)



文字alphaが何番目の  
アルファベットかを調べる

# 文字列(String)

---



```
String test="KANAGAWA";
```

```
test.substring()
```

指定した場所からの  
文字列を取り出す

部分文字列

```
test.substring(4)->"GAWA"
```

```
test.substring(6)->"WA"
```

```
test.substring(0)->"KANAGAWA"
```

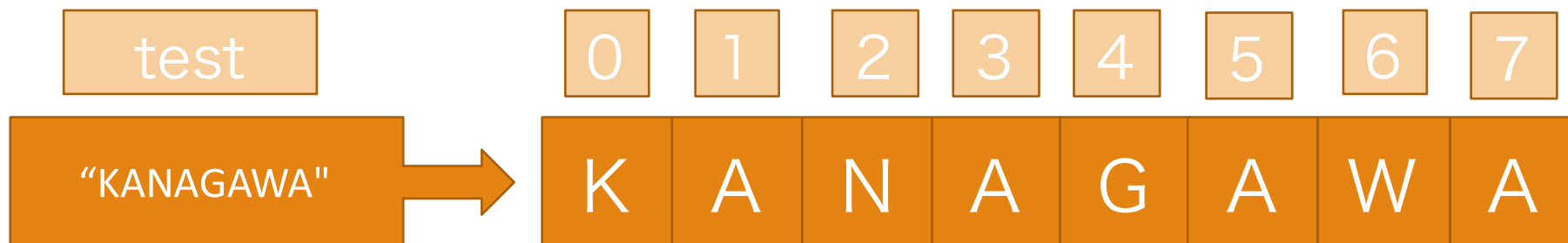
```
test.substring(4,6)->"GA"
```

```
test.substring(1,6)->"ANAGA"
```

```
test.substring(2,5)->"NAG"
```

# 文字列(String)

---



```
String test="KANAGAWA";
```



```
test + " Institute" -> "KANAGAWA Institute "
```

```
test + "2016" -> "KANAGAWA2016"
```

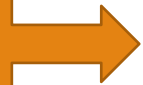
```
test + test -> "KANAGAWAKANAGAWA"
```

```
test + test+ " Institute" -> "KANAGAWA KANAGAWA Institute "
```

# 文字列 (String)

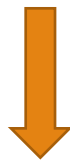
---

split



指定した文字で文字列を分解する

```
String[] moji = "Akagi, aircraft carrier, 1925".split(",")
```



“,”の位置で文字列を分割

```
moji[0] = "Akagi", moji[1] = "aircraft carrier", moji[2] = "1925"
```

# 文字列 (String)

---

trim



文字列中の前後の  
ホワイトスペース（空白など）を取り除く

“ K A ”.trim() -> “K A”

isEmpty()



空の文字列かを判定する

test.isEmpty() -> false

“Kanagawa”.isEmpty() -> false

“”.isEmpty() -> true

# 文字列と数値の変換

---

関数名	機能	使用例
<code>int(String s)</code>	引数が表している数値をint型の数値に変換	<code>int("123")</code>
<code>float(String s)</code>	引数が表している数値をfloat型の数値に変換	<code>float("3.14")</code>
<code>str(int x)</code>	int型引数xを文字列に変換	<code>str(123)</code>
<code>str(float x)</code>	float型引数xを文字列に変換	<code>str(3.14)</code>

# CSVファイル

---

Excelとかテキストエディタで保存出来る形式

拡張子：.csv

# ファイルからのデータの読み込み

---

一気に全てのデータを  
読み出す

読み込んだデータを文字列の  
配列に入れる

一行ずつデータを  
読み出す

一文字ずつデータを  
読み出す



# ファイルからのデータの読み込み

---

一気に全てのデータを読み出す

`String[] loadStrings (ファイル名)`

```
String[] lines = loadString("fuel.txt");
```

# ファイルへのデータの書き出し

---

一気に全てのデータを  
書き出す

一行ずつデータを  
書き出す

一文字ずつデータを  
書き出す

# ファイルへのデータの書き出し

---

一気に全てのデータを書き出す

saveStrings関数

saveStrings(ファイル名,文字列配列)

saveStrings("test.txt",lines)

String[] linesとする

lines変数に入っている文字列を

"test.txt"に書き出す

# 授業時に配布した資料

---

<http://www.sato-lab.jp/imfu/index.html>

においてあります。