TABLES

CS 106 Winter 2021

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your favourites!

Created by: Spotify • 30 songs, 2 hr 36 min

PAUSE FOLLOWING ···

FOLLOWER
1

Filter

Download

	SONG	ARTIST	ALBUM		
+	Ways To Go - Margot Mix	Weval, Margot	Weval Remix	11 hours ago	7:11
+	Death Is A Girl	Mini Mansions	The Great Preten...	11 hours ago	4:36
+	Jumbo	Underworld	Beaucoup Fish	11 hours ago	6:58
+	Bug Powder Dust	The Mysterons	Meandering	11 hours ago	4:27
+	...To Have No Answer	Flock of Dimes	If You See Me, Sa...	11 hours ago	3:49
+	I'll Cut You Down	Uncle Acid & The...	Blood Lust	11 hours ago	5:02
+	L'enfer ce n'est pas les autres c'est moi	The Eye Of Time	Myth I: A Last Da...	11 hours ago	5:46
+	Terrain	pg.lost	Key	11 hours ago	5:29

2:46 3:54

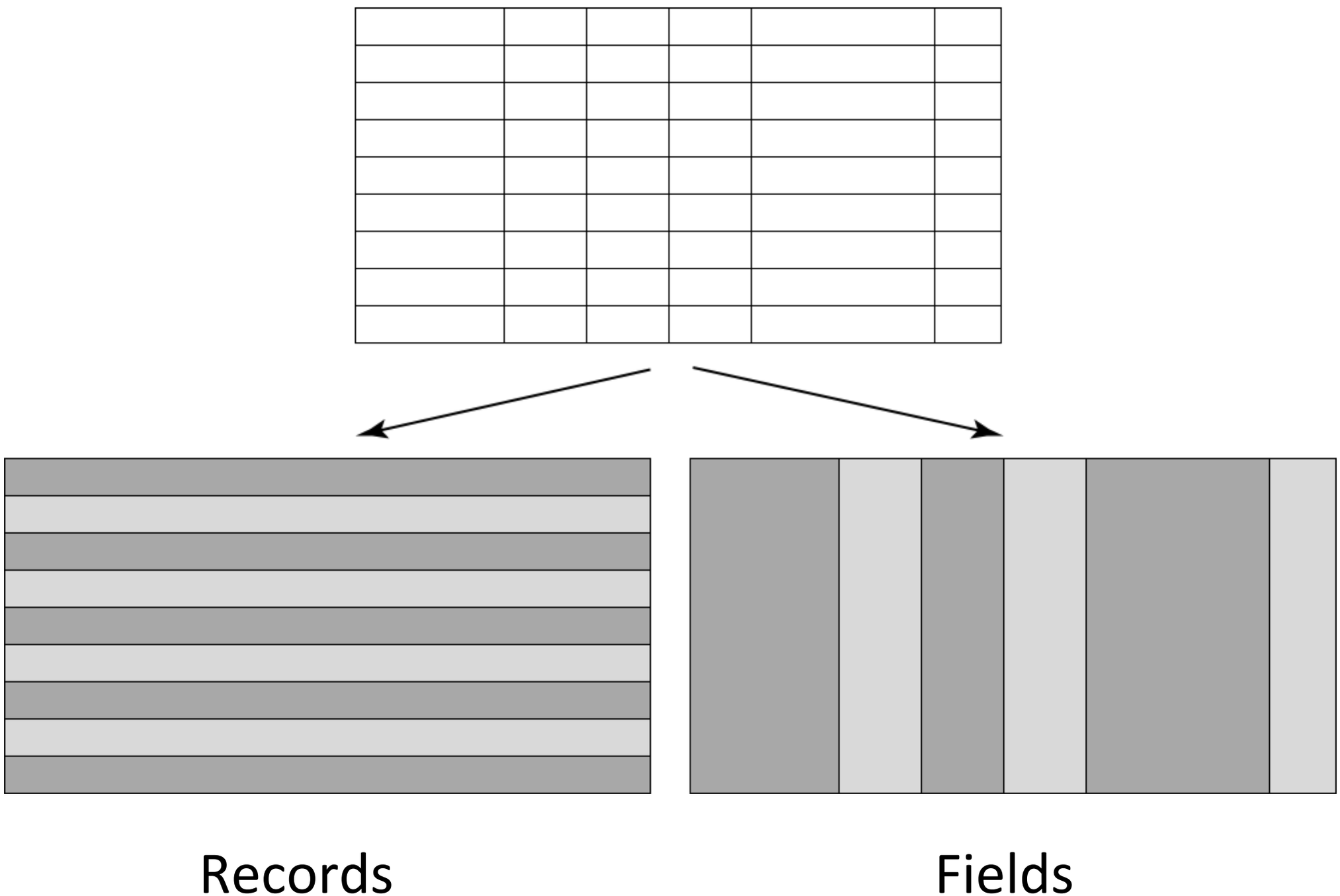
haireyecolour.csv

In Excel: Cells are referenced by column and row (ex: c6 is “blue”);

C6 | Blue

	A	B	C	D	E
1		Hair	Eye	Sex	Freq
2	1	Black	Brown	Male	32
3	2	Brown	Brown	Male	53
4	3	Red	Brown	Male	10
5	4	Blond	Brown	Male	3
6	5	Black	Blue	Male	11
7	6	Brown	Blue	Male	50
8	7	Red	Blue	Male	10
9	8	Blond	Blue	Male	30
10	9	Black	Hazel	Male	10

A table is a grid. Each row is a *record*, and each column is a *field*. We can think of the table as a **sequence** of records.

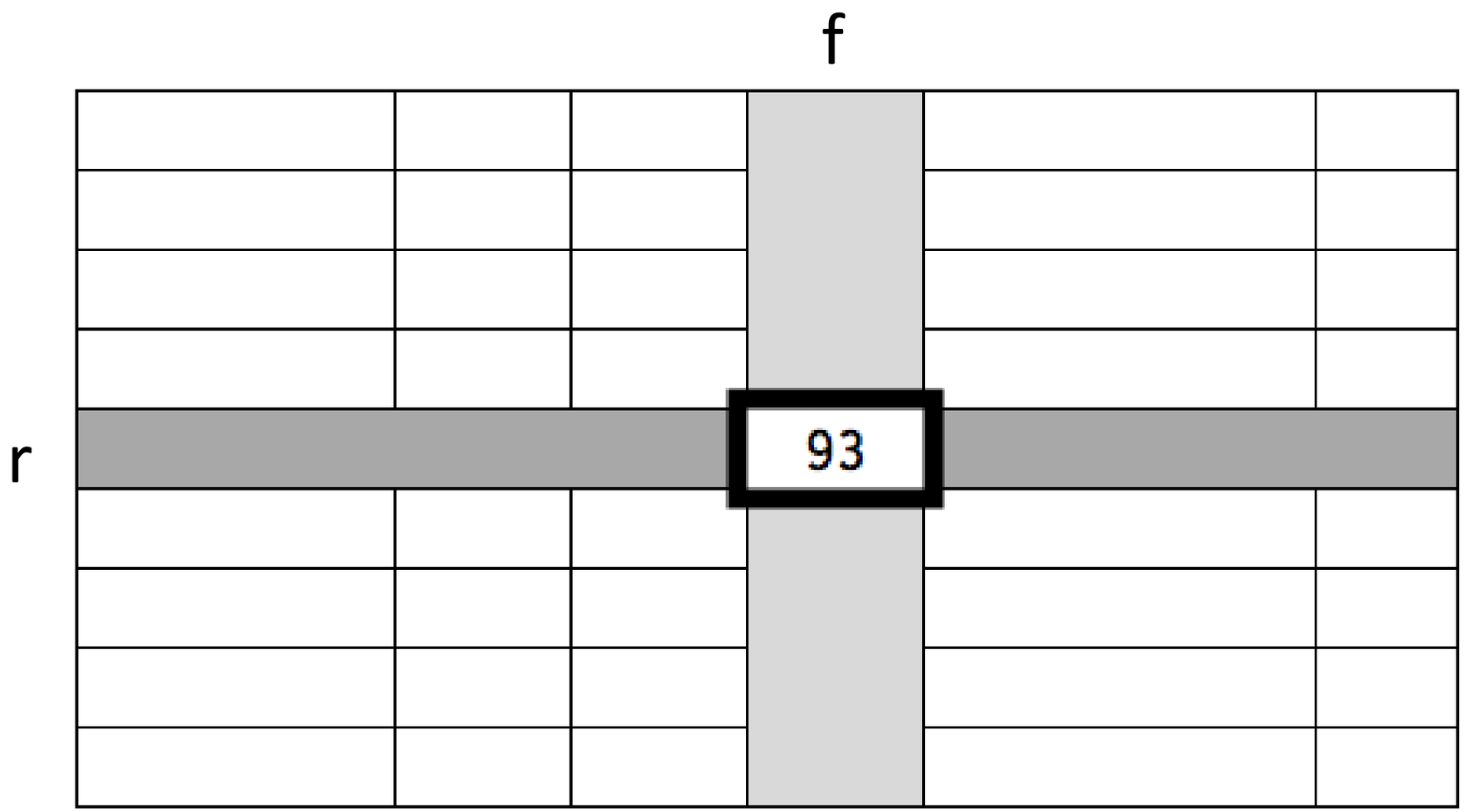


All values in a given field/column have the same type, but different fields can have different types.

1. Column A is of type String (could be Num also)
2. Column B is of type String
3. Column C is of type String
4. Column D is of type String
5. Column E is of type Num

C6 | Blue

	A	B	C	D	E
1		Hair	Eye	Sex	Freq
2	1	Black	Brown	Male	32
3	2	Brown	Brown	Male	53
4	3	Red	Brown	Male	10
5	4	Blond	Brown	Male	3



A *cell* holds one record's value for one field. A cell's location is "two-dimensional"—it takes two values to describe its location.

C6 | Blue

	A	B	C	D	E
1		Hair	Eye	Sex	Freq
2	1	Black	Brown	Male	32
3	2	Brown	Brown	Male	53
4	3	Red	Brown	Male	10
5	4	Blond	Brown	Male	3
6	5	Black	Blue	Male	11
7	6	Brown	Blue	Male	50
8	7	Red	Blue	Male	10
9	8	Blond	Blue	Male	30
10	9	Black	Hazel	Male	10

Some tables have “header rows” that give names to the fields.

Comma-separated values

- CSV: a standard, simple text-based file format for tabular data.
- This is how csv files are stored on disk.

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
```

This is how csv files are displayed by Excel.

C6 | Blue

	A	B	C	D	E
1		Hair	Eye	Sex	Freq
2	1	Black	Brown	Male	32
3	2	Brown	Brown	Male	53
4	3	Red	Brown	Male	10
5	4	Blond	Brown	Male	3
6	5	Black	Blue	Male	11
7	6	Brown	Blue	Male	50
8	7	Red	Blue	Male	10
9	8	Blond	Blue	Male	30
10	9	Black	Hazel	Male	10

Representing Tables

in JavaScript p5

Home
Download
Start
Reference
Libraries
Learn
Examples
Books
Community
Forum
GitHub

Reference

Search the API

p5.Table

Description

`Table` objects store data with multiple rows and columns, much like in a traditional spreadsheet. Tables can be generated from scratch, dynamically, or using data from an existing file.

Syntax

```
new p5.Table([rows])
```

Parameters

`rows` `p5.TableRow`[]: An array of p5.TableRow objects

Loading a table in JavaScript p5

```
function preload() {
  tab = loadTable("/data/haireyecolour.csv", "header");
}
```

Read CSV data from a file, create a variable to store it.

Note that the first row is treated differently than all other rows, as it is the “header” row.

https://openprocessing.org/sketch/1122348

Column names (i.e. header row) are stored in an array .columns

```
function setup() {
  print("Column 0 name:", tab.columns[0]);
  print("Column 1 name:", tab.columns[1]);
  print("Column 2 name:", tab.columns[2]);
  print("Column 3 name:", tab.columns[3]);
  print("Column 4 name:", tab.columns[4]);
}
```

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
```

```
Column 0 name:
Column 1 name: Hair
Column 2 name: Eye
Column 3 name: Sex
Column 4 name: Freq
```

We can use a loop to iterate over the header row

```
function setup() {
  for (let col = 0; col < 5; col++) {
    print("Column ", col, "name:", tab.columns[col]);
  }
}
```

https://openprocessing.org/sketch/1122359

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
```

```
Column  0 name:
Column  1 name: Hair
Column  2 name: Eye
Column  3 name: Sex
Column  4 name: Freq
```

Number of Rows and Columns

```
tab.getRowCount();
tab.getColumnCount();
```

Controlling the loop with `tab.getColumnCount`

```
function setup() {
  for (let col = 0; col < tab.getColumnCount(); col++) {
    let fieldName = tab.columns[col];
    print("Column", col, "name:", fieldName);
  }
}
```

https://openprocessing.org/sketch/1122361

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
```

```
Column 0 name:
Column 1 name: Hair
Column 2 name: Eye
Column 3 name: Sex
Column 4 name: Freq
```

The same code as the previous slide except result is displayed on the canvas (i.e. using text() rather than print())

```
function setup() {
  createCanvas(500, 100);
  background(220);
  textSize(24);
  for (let col = 0; col < tab.getColumnCount(); col++) {
    let fieldName = tab.columns[col];
    text(fieldName, col * 100 + 10, 30);
  }
}
```

https://openprocessing.org/sketch/1122363

Reading cells in JavaScript p5

To read a cell value, you need to know **three** things about it:

1. The record: what **row** is the cell in?
2. The field: what **column** is the cell in?
3. What **type** of data do you expect to find there?

1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25

Column

Row	0	1	2	3	4
0	1	Black	Brown	Male	32
1	2	Brown	Brown	Male	53
2	3	Red	Brown	Male	10
3	4	Blond	Brown	Male	3
4	5	Black	Blue	Male	11
5	6	Brown	Blue	Male	50
6	7	Red	Blue	Male	10
7	8	Blond	Blue	Male	30
8	9	Black	Hazel	Male	10
9	10	Brown	Hazel	Male	25

```
let freq = tab.getNum(6, 4);
```

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
"8","Blond","Blue","Male",30
"9","Black","Hazel","Male",10
"10","Brown","Hazel","Male",25
```

Header row

Column

	0	1	2	3	4
		Hair	Eye	Sex	Freq
0	1	Black	Brown	Male	32
1	2	Brown	Brown	Male	53
2	3	Red	Brown	Male	10
3	4	Blond	Brown	Male	3
4	5	Black	Blue	Male	11
5	6	Brown	Blue	Male	50
6	7	Red	Blue	Male	10
7	8	Blond	Blue	Male	30
8	9	Black	Hazel	Male	10
9	10	Brown	Hazel	Male	25

Row

Column

Row	0	1	2	3	4
		Hair	Eye	Sex	Freq
0	1	Black	Brown	Male	32
1	2	Brown	Brown	Male	53
2	3	Red	Brown	Male	10
3	4	Blond	Brown	Male	3
4	5	Black	Blue	Male	11
5	6	Brown	Blue	Male	50
6	7	Red	Blue	Male	10
7	8	Blond	Blue	Male	30
8	9	Black	Hazel	Male	10
9	10	Brown	Hazel	Male	25

Column

Row

	0	1	2	3	4
		Hair	Eye	Sex	Freq
0	1	Black	Brown	Male	32
1	2	Brown	Brown	Male	53
2	3	Red	Brown	Male	10
3	4	Blond	Brown	Male	3
4	5	Black	Blue	Male	11
5	6	Brown	Blue	Male	50
6	7	Red	Blue	Male	10
7	8	Blond	Blue	Male	30
8	9	Black	Hazel	Male	10
9	10	Brown	Hazel	Male	25

```
let eyes = tab.getString(4, 2);
```

Column

Row	0	1	2	3	4
		Hair	Eye	Sex	Freq
0	1	Black	Brown	Male	32
1	2	Brown	Brown	Male	53
2	3	Red	Brown	Male	10
3	4	Blond	Brown	Male	3
4	5	Black	Blue	Male	11
5	6	Brown	Blue	Male	50
6	7	Red	Blue	Male	10
7	8	Blond	Blue	Male	30
8	9	Black	Hazel	Male	10
9	10	Brown	Hazel	Male	25

```
let eyes = tab.getString(4, "Eye");
```

Summary of cell Reference

```
// both refer to 53.
someNum = tab.getNum(1, 4);
someNum = tab.getNum(1, "Freq");

// Both refer to "Male".
someString = tab.getString(4, 3);
someString = tab.getString(4, "Sex");
```

Modifying cells

```
function setup() {
  tab.set( 5, 4, 100 );
  tab.set( 5, "Freq", 100 );

  tab.set( 5, 3, "unknown");
  tab.set( 5, "Sex", "unknown" );
}
```

Let’s look at haireyecolour.csv in a bit more detail

- Has a header (index 0-4)
- Has 5 columns/fields (index 0-4)
- Has 32 rows/records (index 0-31)
- haireyecolour.csv is students in a class. For example:
  - 32 are male with black hair and brown eyes
  - 53 are male with brown hair and brown eyes
  - In the last row of the table we see
    8 are female with blond hair and green eyes.

```
"","Hair","Eye","Sex","Freq"
"1","Black","Brown","Male",32
"2","Brown","Brown","Male",53
"3","Red","Brown","Male",10
"4","Blond","Brown","Male",3
"5","Black","Blue","Male",11
"6","Brown","Blue","Male",50
"7","Red","Blue","Male",10
"8","Blond","Blue","Male",30
"9","Black","Hazel","Male",10
"10","Brown","Hazel","Male",25
"11","Red","Hazel","Male",7
"12","Blond","Hazel","Male",5
"13","Black","Green","Male",3
"14","Brown","Green","Male",15
"15","Red","Green","Male",7
"16","Blond","Green","Male",8
"17","Black","Brown","Female",36
"18","Brown","Brown","Female",66
"19","Red","Brown","Female",16
"20","Blond","Brown","Female",4
"21","Black","Blue","Female",9
"22","Brown","Blue","Female",34
"23","Red","Blue","Female",7
"24","Blond","Blue","Female",64
"25","Black","Hazel","Female",5
"26","Brown","Hazel","Female",29
"27","Red","Hazel","Female",7
"28","Blond","Hazel","Female",5
"29","Black","Green","Female",2
"30","Brown","Green","Female",14
"31","Red","Green","Female",7
"32","Blond","Green","Female",8
```

How many students in haireyecolour.csv

```
let table;
function preload() {
  table = loadTable( "/data/haireyecolour.csv", "header" );
}
function setup() {
  let total = 0;
  for (let row=0;row < table.getRowCount();row++) {
    total += table.getNum( row, "Freq" );
  }
  print("total number of students is:", total);
}
```

https://openprocessing.org/sketch/1122374

```
total number of students is: 592
```

Demo Code

on Open Processing
for the following example

“HairEyeColourSpreadsheetP5”

https://openprocessing.org/sketch/1122393

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8
17	Black	Brown	Female	36
18	Brown	Brown	Female	66
19	Red	Brown	Female	16

```
let startRow;

function preload() {
  tab = loadTable("/data/haireyecolour.csv", "header");
}

function setup() {
  createCanvas(800, 800);
  startRow = 0;
  textSize(24);
}
```

```
function draw() {
  background(0);

  // Draw the header row.
  fill(255);
  noStroke();
  rect(0, 0, width, 40);
  fill(0);
for (let col = 0; col < tab.getColumnCount(); col++) {
    let fieldName = tab.columns[col];
    text(fieldName, 10 + col * 150, 10  + 24);
  }
```

```
// Draw the rest of the table.
 fill(255);
 for (let row = 0; row < height / 40 + 1; row++) {
   if ((row + startRow) < tab.getRowCount()) {
     for (let col = 0; col < tab.getColumnCount(); col++) {
       let cell = tab.get(row + startRow, col);
       text(cell, 10 + col * 150, 10 + (row + 1) * 40 + 24);
     }
   }
 }
}
```

```
function keyPressed() {
  if (keyCode === DOWN_ARROW) {
    startRow++;
  } else if (keyCode === UP_ARROW) {
    startRow = max(0, startRow - 1);
  }
}
```

Baseball salaries

Two CSV files

- Master.csv
- Salaries.csv

seanlahman.com/baseball-archive/statistics/

Master.csv

- **Header record**
- **One record/row per player**
- **19,106 rows/records**
- **24 columns/fields ()**
- **Unique playerID**

```
playerID,birthYear,birthMonth,birthDay,birthCountry,birthState,birthCity,deathYear,deathMonth,
aardsda01,1981,12,27,USA,CO,Denver,,,,,,,David,Aardsma,David Allan,215,75,R,R,2004-04-06,2015
aaronha01,1934,2,5,USA,AL,Mobile,,,,,,,Hank,Aaron,Henry Louis,180,72,R,R,1954-04-13,1976-10-0
aaronto01,1939,8,5,USA,AL,Mobile,1984,8,16,USA,GA,Atlanta,Tommie,Aaron,Tommie Lee,190,75,
aasedo01,1954,9,8,USA,CA,Orange,,,,,,,Don,Aase,Donald William,190,75,R,R,1977-07-26,1990-10-0
abadan01,1972,8,25,USA,FL,Palm Beach,,,,,,,Andy,Abad,Fausto Andres,184,73,L,L,2001-09-10,2006
abadfe01,1985,12,17,D.R.,La Romana,La Romana,,,,,,,Fernando,Abad,Fernando Antonio,220,73,L,L
abadijo01,1850,11,4,USA,PA,Philadelphia,1905,5,17,USA,NJ,Pemberton,John,Abadie,John W.,192,
abbated01,1877,4,15,USA,PA,Latrobe,1957,1,6,USA,FL,Fort Lauderdale,Ed,Abbaticchio,Edward Jar
abbeybe01,1869,11,11,USA,VT,Essex,1962,6,11,USA,VT,Colchester,Bert,Abbey,Bert Wood,175,71,
abbeych01,1866,10,14,USA,NE,Falls City,1926,4,27,USA,CA,San Francisco,Charlie,Abbey,Charles S.
abbotda01,1862,3,16,USA,OH,Portage,1930,2,13,USA,MI,Ottawa Lake,Dan,Abbott,Leander Frankli
```

Master.csv loaded into Excel.
(19,106 rows and 24 columns)

	A	B	C	D	E
1	playerID	birthYear	birthMont	birthDay	birthCoun
2	aardsda01	1981	12	27	USA
3	aaronha01	1934	2	5	USA
4	aaronto01	1939	8	5	USA
5	aasedo01	1954	9	8	USA
6	abadan01	1972	8	25	USA
7	abadfe01	1985	12	17	D.R.
8	abadijo01	1850	11	4	USA
9	abbated01	1877	4	15	USA
10	abbeybe0	1869	11	11	USA
11	abbeych01	1866	10	14	USA
12	abbotda01	1862	3	16	USA

```
// A simple sketch to display the first ten baseball
//  players in Master.csv.
let master = [];
let myDiv;
let myText = "";

function preload() {
  master=loadTable( "/data/Master.csv", "header" );
}
```

https://openprocessing.org/sketch/1122404

```
function setup(){
  noCanvas();
  myDiv = createDiv();
  myDiv.style("font-size", "48");

  for ( let row = 0; row < 10; row++ ) {
    myText = myText + "<br>" +
      master.getString( row, "playerID" ) + "    " +
      master.getString( row, "nameFirst" ) + "    " +
      master.getString( row, "nameLast" );
  }
  myDiv.html(myText);
}
}
```

aardsda01 David Aardsma
aaronha01 Hank Aaron
aaronto01 Tommie Aaron
aasedo01 Don Aase
abadan01 Andy Abad
abadfe01 Fernando Abad
abadijo01 John Abadie
abbated01 Ed Abbaticchio
abbeybe01 Bert Abbey
abbeych01 Charlie Abbey

The above code is in the demo code:

“BaseballMasterDisplayedP5”

Baseball salaries

Two CSV files

- Master.csv
- Salaries.csv

seanlahman.com/baseball-archive/statistics/

Salaries.csv

- Header record
- It has one record per player per year
    - i.e. same playerID occurs multiple times
- 26,429 rows/records
- 5 columns/fields

```
yearID,teamID,lgID,playerID,salary
1985,ATL,NL,barkele01,870000
1985,ATL,NL,bedrost01,550000
1985,ATL,NL,benedbr01,545000
1985,ATL,NL,campri01,633333
1985,ATL,NL,ceronri01,625000
1985,ATL,NL,chambch01,800000
```

seanlahman.com/baseball-archive/statistics/

The same CSV File of Baseball Salaries loaded into Excel. (26,429 rows and four columns)

	A	B	C	D	E	F
1	yearID	teamID	lgID	playerID	salary	
2	1985	ATL	NL	barkele01	870000	
3	1985	ATL	NL	bedrost01	550000	
4	1985	ATL	NL	benedbr0	545000	
5	1985	ATL	NL	campri01	633333	
6	1985	ATL	NL	ceronri01	625000	
7	1985	ATL	NL	chambch0	800000	
8	1985	ATL	NL	dedmoje0	150000	
9	1985	ATL	NL	forstte01	483333	
10	1985	ATL	NL	garbege01	772000	
11	1985	ATL	NL	harpete01	250000	
12	1985	ATL	NL	horneho0	1500000	

```
// A simple sketch to display the first
//   ten records in Salaries.csv.
let sals = [];
let myDiv;
let myText = "";

function preload() {
  sals = loadTable( "/data/Salaries.csv", "header" );
}
```

https://openprocessing.org/sketch/1122411

```
function setup(){
noCanvas();
  myDiv = createDiv();
  myDiv.style("font-size", "48");
  for ( let row = 0; row < 10; row++ ) {
    myText = myText + "<br>" +
      sals.getString( row, "yearID" ) + "    " +
      sals.getString( row, "teamID" ) + "    " +
      sals.getString( row, "lgID" ) + "    " +
      sals.getString( row, "playerID" ) + "    " +
      sals.getString( row, "salary" );
  }
myDiv.html(myText);
}
```

```
1985 ATL NL barkele01 870000
1985 ATL NL bedrost01 550000
1985 ATL NL benedbr01 545000
1985 ATL NL campri01 633333
1985 ATL NL ceronri01 625000
1985 ATL NL chambch01 800000
1985 ATL NL dedmoje01 150000
1985 ATL NL forstte01 483333
1985 ATL NL garbege01 772000
1985 ATL NL harpete01 250000
```

The above code is in the demo code:

“BaseballSalariesDisplayedP5”

Master.csv and Salaries.csv: Combined

- List names and salary of all players in all years, who make the maximum salary
- Need both tables
  - Player’s name is in Master.csv
  - Player salary’s are in Salaries.csv

Algorithm

- Iterate over Master.csv and create a dictionary, and store the players in it
- Iterative over Salaries.csv to find max salary
- Iterated over Salaries.csv again to:
  - Find each player with the maximum salary
    - Save year from Salaries.csv
    - Get and save their name from players dictionary
- Display the results

```
let player_dict = {};
let master;
let sals;
let myDiv;

function preload() {
  master = loadTable( "Master.csv", "header" );
  sals = loadTable( "Salaries.csv", "header" );
}
```

https://openprocessing.org/sketch/1122422

```
function setup(){

noCanvas();

  myDiv = createDiv();

  myDiv.style("font-size", "48");

// Create a dictionary and store the players in it.

  for ( let row = 0; row < master.getRowCount(); row++ ) {

  // Associate the given key with the player's full name.

  let playerID = master.getString( row, "playerID" );

  player_dict[playerID] = master.getString( row, "nameFirst") +

       " " + master.getString( row, "nameLast" );

}
```

```
// Find the maximum salary. This is in the sals table.
let maxSalary = sals.getString( 0, "salary" );
for ( let row = 1; row < sals.getRowCount(); row++ ) {
  let tempSalary =   sals.getNum( row, "salary" );
  if (tempSalary > maxSalary) {
    maxSalary = sals.getNum( row, "salary" );
  }
}
```

```
  // Find all players who have the maximum salary.
let myText = "";
  for ( let row = 0; row < sals.getRowCount(); row++ ) {
    let tempSalary =   sals.getNum( row, "salary" );
    if (tempSalary === maxSalary) {
      idOfPlayerWithMaxSalary = sals.getString( row, "playerID" );
      maxSalaryYear = sals.getString(row, "yearID");
      let name = player_dict[idOfPlayerWithMaxSalary];
      myText = myText + "<br><br>" +
          name + "<br>" +
          maxSalaryYear +
          "<br>" + "$" + nfc(maxSalary, 0);
    }
  }

// Display the result on the DOM

  myDiv.html(myText);

}
```

Alex Rodriguez
2009
$33,000,000

Alex Rodriguez
2010
$33,000,000

Clayton Kershaw
2016
$33,000,000

The above code is in the demo code:

“BaseballSalariesP5”

W21

- In the Winter 2021 semester the following slides are not covered. The following slides show examples of combining 3 csv files. This was covered in previous years but is not covered in Winter 2021.

- Combining 3 csv files will not be on any lab, assignment, or test in W2021.

- The following slides are included only for the benefit of any students who would like to look at the examples and see how 3 csv files are combined.

Another demo code example:

Fully implemented sketch with three files being combined.

It is about food inspection in Waterloo Region.

“FoodInspectionsP5”

https://openprocessing.org/sketch/1123932

This slide is not required for CS106 W21.

Regional food inspections

Facilities_OpenData.csv

“FACILITYID","BUSINESS_NAME","TELEPHONE","ADDR","CITY","EATSMART","OPEN_DATE","DESCRIPT
"B5AB474B-2CBC-4D61-B100-670BB6EE6AD7","YE'S SUSHI","519-888-6066","B8 - 583 KING ST N","W
"CCA5C401-01EF-42AE-832C-7AB24C201263","KISMET RESTAURANT","(519) 746-8788","20 - 160 UNI
…

Inspections_OpenData.csv

"INSPECTION_ID","FACILITYID","INSPECTION_DATE","REQUIRE_REINSPECTION","CERTIFIED_FOOD_HA
"{56D1AB86-5392-452E-8336-000964689795}","081F0F8A-892E-41F7-811C-9CEE8D690A14","2016/0
"{67DF7158-C081-412B-B6DC-000C251E98F6}","23AA5EBA-35C8-47FB-8C95-198C50C72B92","2015/1
"{C449B882-89A0-4F47-B05A-000C326432A1}","F1CFD836-8A73-4A03-85E0-1EE669470E26","2015/0
…

Infractions_OpenData.csv

"INFRACTION_ID","INSPECTION_ID","INFRACTION_TYPE","Infr ctio
"{187D230D-954E-426E-AD29-2622155F4C16}","{C45B2081-E ON-
"{5D5A9FDC-019D-4275-A6E0-6C4E97DD136A}","{C45B2081-E NON
…

This slide is not required for CS106 W21.

www.regionofwaterloo.ca/en/regionalGovernment/FoodPremiseDataset.asp

Regional food inspections

1. Get restaurant name from user.
2. Look up corresponding FACILITYID in Facilities_OpenData.csv.
3. In Inspections_OpenData.csv, find all INSPECTION_IDs that have the same FACILITYID.
4. In Infractions_OpenData.csv, find all INFRACTION_IDs associated with any of these INSPECTION_IDs.
5. Report the text of the infractions.

This slide is not required for CS106 W21.

www.regionofwaterloo.ca/en/regionalGovernment/FoodPremiseDataset.asp