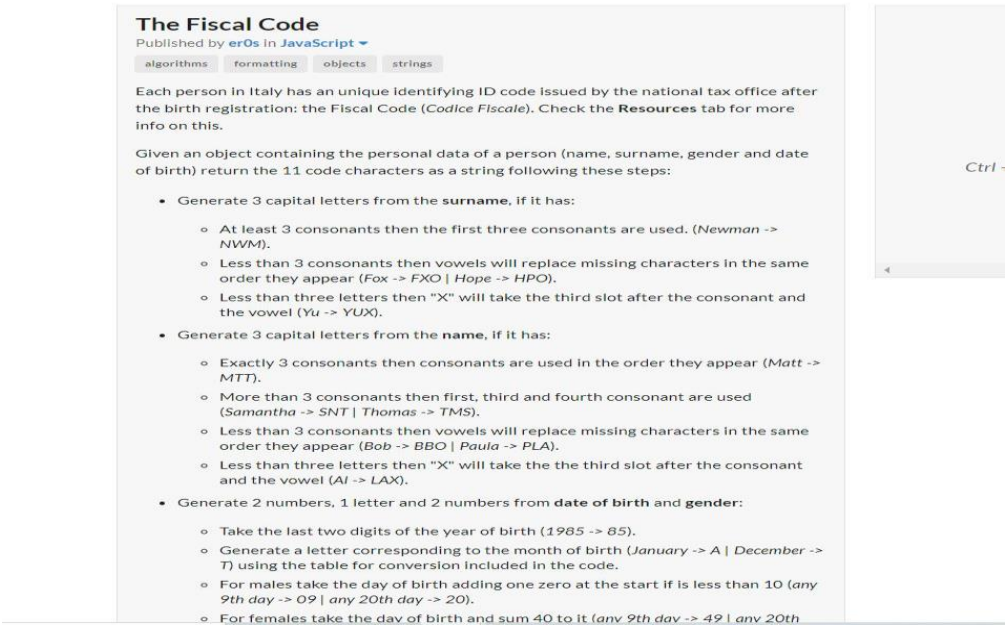
ข้อ 1



The Fiscal Code

Published by **er0s** in **JavaScript**

algorithms formatting objects strings

Each person in Italy has an unique identifying ID code issued by the national tax office after the birth registration: the Fiscal Code (*Codice Fiscale*). Check the **Resources** tab for more info on this.

Given an object containing the personal data of a person (name, surname, gender and date of birth) return the 11 code characters as a string following these steps:

- Generate 3 capital letters from the **surname**, if it has:
  - At least 3 consonants then the first three consonants are used. (*Newman -> NWM*).
  - Less than 3 consonants then vowels will replace missing characters in the same order they appear (*Fox -> FXO | Hope -> HPO*).
  - Less than three letters then "X" will take the third slot after the consonant and the vowel (*Yu -> YUX*).
- Generate 3 capital letters from the **name**, if it has:
  - Exactly 3 consonants then consonants are used in the order they appear (*Matt -> MTT*).
  - More than 3 consonants then first, third and fourth consonant are used (*Samantha -> SNT | Thomas -> TMS*).
  - Less than 3 consonants then vowels will replace missing characters in the same order they appear (*Bob -> BBO | Paula -> PLA*).
  - Less than three letters then "X" will take the the third slot after the consonant and the vowel (*Al -> LAX*).
- Generate 2 numbers, 1 letter and 2 numbers from **date of birth** and **gender**:
  - Take the last two digits of the year of birth (*1985 -> 85*).
  - Generate a letter corresponding to the month of birth (*January -> A | December -> T*) using the table for conversion included in the code.
  - For males take the day of birth adding one zero at the start if is less than 10 (*any 9th day -> 09 | any 20th day -> 20*).
  - For females take the day of birth and sum 40 to it (*any 9th day -> 49 | any 20th*

```
fiscalCode({
  name: "Matt",
  surname: "Edabit",
  gender: "M",
  dob: "1/1/1900"
}) → "DBTMTT00A01"

fiscalCode({
  name: "Helen",
  surname: "Yu",
  gender: "F",
  dob: "1/12/1950"
}) → "YUXHLN50T41"

fiscalCode({
  name: "Mickey",
  surname: "Mouse",
  gender: "M",
  dob: "16/1/1928"
}) → "MSOMKY28A16"
```

Code

1TheFiscalCode.js

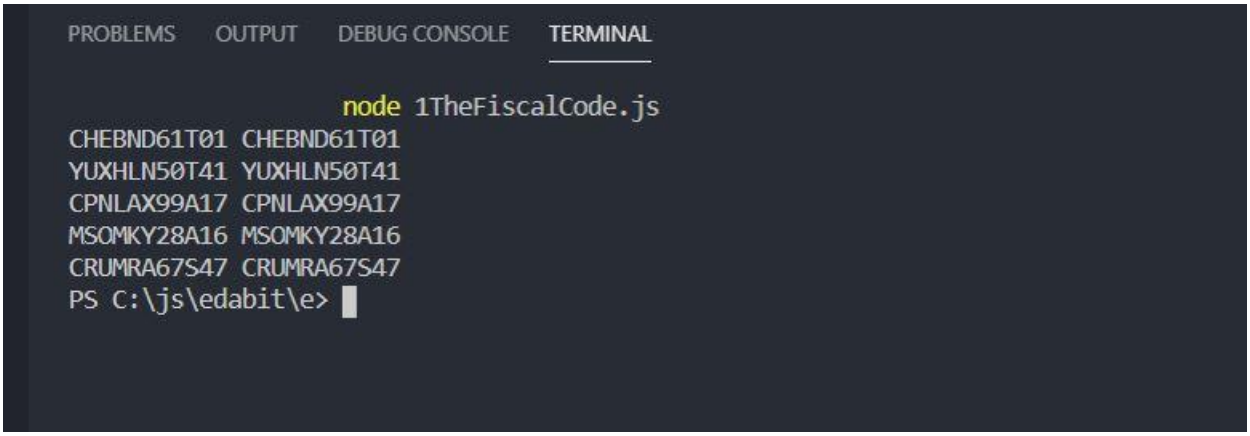
```
const months = {
    1: 'A',
    2: 'B',
    3: 'C',
    4: 'D',
    5: 'E',
    6: 'H',
    7: 'L',
    8: 'M',
    9: 'P',
    10: 'R',
    11: 'S',
    12: 'T',
  };

  const genderCode = (gender, dob) => {
    const [day, month, year] = dob.split('/');
    if (gender === 'M') {
      return +day < 10 ? '0' + day : day;
    } else {
      return +day + 40;
    }
  };

  const monthCode = (dob) => {
    const [day, month, year] = dob.split('/');
    return months[month];
  };
```

```
const genderCode = (gender, dob) => {
  const [day, month, year] = dob.split('/');
  if (gender === 'M') {
    return +day < 10 ? '0' + day : day;
  } else {
    return +day + 40;
  }
};

const monthCode = (dob) => {
  const [day, month, year] = dob.split('/');
  return months[month];
};

const yearCode = (dob) => {
  return dob.slice(-2);
};

const surnameCode = (name) => {
  let vowels = name.match(/[aeiou]/gi);
  let consonants = name.match(/[^aeiou]/gi);
  let result;

  if (consonants.length > 2) {
    result = consonants;
  } else if ([...consonants, ...vowels].length < 3) {
    result = [
      ...consonants,
      ...vowels,
      'X'.repeat(3 - [...consonants, ...vowels].length),
    ];
  } else if (consonants.length < 3) {
    result = [...consonants, ...vowels];
  }
```

```
50
51      return result.slice(0, 3).join('').toUpperCase();
52    };
53
54    const nameCode = (name) => {
55      let vowels = name.match(/[aeiou]/gi);
56      let consonants = name.match(/[^aeiou]/gi);
57      let result = [];
58
59      if (consonants.length === 3) {
60        result = consonants;
61      } else if (consonants.length > 3) {
62        result = [consonants[0], consonants[2], consonants[3]];
63      } else if ([...consonants, ...vowels].length < 3) {
64        result = [
65          ...consonants,
66          ...vowels,
67          'X'.repeat(3 - [...consonants, ...vowels].length),
68        ];
69      } else if (consonants.length < 3) {
70        result = [...consonants, ...vowels];
71      }
72
73      return result.slice(0, 3).join('').toUpperCase();
74    };
75
76    const fiscalCode = (person) => {
77      const { name, surname, gender, dob } = person;
78      return (
79        surnameCode(surname) +
80        nameCode(name) +
81        yearCode(dob) +
82        monthCode(dob) +
83        genderCode(gender, dob)
84      );
85    };
86
```

```
  console.log(
    fiscalCode({
      name: 'Brendan',
      surname: 'Eich',
      gender: 'M',
      dob: '1/12/1961',
    }),
    'CHEBND61T01'
  );
  console.log(
    fiscalCode({ name: 'Helen', surname: 'Yu', gender: 'F', dob: '1/12/1950' }),
    'YUXHLN50T41'
  );
  console.log(
    fiscalCode({ name: 'Al', surname: 'Capone', gender: 'M', dob: '17/1/1899' }),
    'CPNLAX99A17'
  );
  console.log(
    fiscalCode({
      name: 'Mickey',
      surname: 'Mouse',
      gender: 'M',
      dob: '16/1/1928',
    }),
    'MSOMKY28A16'
  );
  console.log(
    fiscalCode({
      name: 'Marie',
      surname: 'Curie',
      gender: 'F',
      dob: '7/11/1867',
    }),
    'CRUMRA67S47'
  );
```

Run


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
node 1TheFiscalCode.js
CHEBND61T01 CHEBND61T01
YUXHLN50T41 YUXHLN50T41
CPNLAX99A17 CPNLAX99A17
MSOMKY28A16 MSOMKY28A16
CRUMRA67S47 CRUMRA67S47
PS C:\js\edabit\e>