

```
FILE *fopen(char *filename, char *mode)
```

Open een bestand (H4, slide 8)

```
int fclose(FILE *fp)
```

Sluit een bestand (H4, slide 9)

```
int fgetc(FILE *fp)
```

Lees karakter in uit bestand (H4, slide 9)

```
int fputc(char c, FILE *fp)
```

Schrijf karakter naar bestand (H4, slide 9)

```
char *fgets(char *line, int maxline, FILE *fp);
```

Lees lijn in (H4, slide 19)

```
int fputs(char *string, FILE *fp)
```

Schrijf string naar bestand (H4, slide 19)

```
int fprintf/fscanf(FILE *fp, char *format, ...)
```

Schrijf/lees formaat naar/uit bestand (H4, slide 14)

```
FILE *fopen(char *filename, char *mode)
```

Open een bestand (H4, slide 9)

```
FILE *file = fopen("bestand.txt", "r");
```

```
FILE *file = fopen("/pad/naar/bestand/bestand.txt", "w");
```

```
int fclose(FILE *fp)
```

Sluit een bestand (H4, slide 10)

```
fclose(file);
```

```
Dit is lijn 1.  
Dit is lijn 2.  
Dit is lijn 3.  
Dit is lijn 4.  
Dit is lijn 5.  
Dit is lijn 6.  
Dit is lijn 7.  
Dit is lijn 8.  
Dit is lijn 9.  
Dit is lijn 10.  
Dit is lijn 11.  
Dit is lijn 12.  
Dit is lijn 13.  
Dit is lijn 14.  
Dit is lijn 15.|
```

```
FILE *file = fopen("bestand.txt", "r");  
  
int fgetc(FILE *fp)  
    Lees karakter in uit bestand (H4, slide 10)  
  
    char c = fgetc(file); // 'D'  
    fgetc(file); // 'i'  
    fgetc(file); // 't'  
    fgetc(file); // ' '  
  
    ...  
    fgetc(file); // EOF
```

```
FILE *file = fopen("bestand.txt", "w");  
int fputc(char c, FILE *fp)  
    Schrijf karakter naar bestand (H4, slide 10)  
    Returnt EOF als er een fout is opgetreden
```

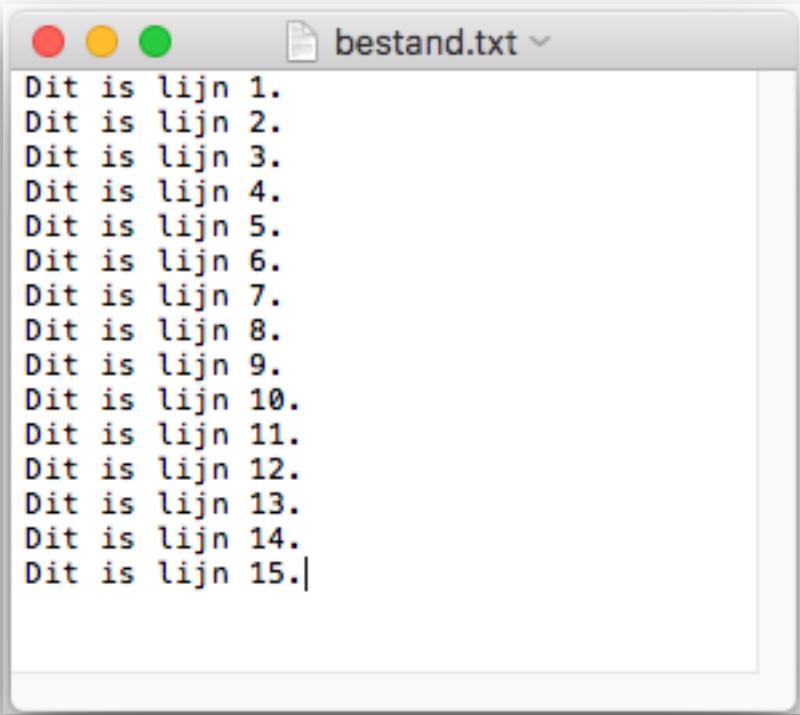
```
fputc('A', file); // Schrijf 'A'  
fputc('b', file); // Schrijf 'b'
```

```
Dit is lijn 1.  
Dit is lijn 2.  
Dit is lijn 3.  
Dit is lijn 4.  
Dit is lijn 5.  
Dit is lijn 6.  
Dit is lijn 7.  
Dit is lijn 8.  
Dit is lijn 9.  
Dit is lijn 10.  
Dit is lijn 11.  
Dit is lijn 12.  
Dit is lijn 13.  
Dit is lijn 14.  
Dit is lijn 15.
```

```
FILE *file = fopen("bestand.txt", "r");  
  
char *fgets(char *line, int maxline, FILE *fp);  
    Lees lijn in (H4, slide 19)  
    Returnt NULL als file volledig is ingelezen  
  
char line[1000];  
fgets(line, 1000, file); // line == "Dit is lijn 1."
```

```
FILE *file = fopen("bestand.txt", "w");  
int fputs(char *str, FILE *fp)  
Schrijf string naar bestand (H4, slide 19)  
Returnt EOF als er een fout is opgetreden
```

```
fputs("Een string", file); // Schrijf "Een string"
```



```
Dit is lijn 1.
Dit is lijn 2.
Dit is lijn 3.
Dit is lijn 4.
Dit is lijn 5.
Dit is lijn 6.
Dit is lijn 7.
Dit is lijn 8.
Dit is lijn 9.
Dit is lijn 10.
Dit is lijn 11.
Dit is lijn 12.
Dit is lijn 13.
Dit is lijn 14.
Dit is lijn 15.
```

```
FILE *file = fopen("bestand.txt", "w");

int fprintf(FILE *fp, char *format, ...);
Schrijf formaat naar bestand (H4, slide 15)
Returnt aantal karakters weggeschreven

fprintf(file, "%s %s %i\n", "Dit is", "lijn", 1);
// Schrijf "Dit is lijn 1"
```

```
int fscanf(FILE *fp, char *format, ...)
Lees formaat uit bestand (H4, slide 15)
Returnt aantal argumenten dat gematcht werd
```

```
char str1[100], str2[100], str3[100];
int i;
fscanf(file, "%s %s %s %i", str1, str2, str3, &i);
// str1 == "Dit", str2 == "is", str3 == "lijn", i == 1
```

```
fscanf(file, "%i", &i);
"Dit is lijn 1." kan niet gematcht worden met een integer ->
fscanf returnt 0
```